



I N T E R W O V E N

TeamSite® Administration Guide

Release 5.5.1

for Windows NT® and Windows® 2000

© 1999-2002 Interwoven, Inc. All rights reserved.

No part of this publication (hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Interwoven.

Information in this manual is furnished under license by Interwoven, Inc. and may only be used in accordance with the terms of the license agreement. If this software or documentation directs you to copy materials, you must first have permission from the copyright owner of the materials to avoid violating the law which could result in damages or other remedies.

Interwoven, TeamSite, OpenDeploy and the Interwoven logo are trademarks of Interwoven, Inc., which may be registered in certain jurisdictions.

SmartContext, DataDeploy, the tagline and service mark are trademarks of Interwoven, Inc. which may be registered in certain jurisdictions. Windows and Windows NT are registered trademarks of Microsoft Corporation. All other trademarks are owned by their respective owners.

This Interwoven product utilizes third party components under the following copyrights with all rights reserved: Copyright 1995-1999, The Apache Group (www.apache.org); Copyright 1986-1993, 1998, Thomas Williams, Colin Kelley. If you are interested in using these components for other purposes, contact the appropriate vendor.



I N T E R W O V E N

Interwoven, Inc.

803 11th Ave.

Sunnyvale, CA 94089

<http://www.interwoven.com>

Printed in the United States of America

Release 5.5.1

Part # 10-00-10-11-00-551-300

Table of Contents

About This Book 11

- Notation Conventions 11
- Path Name Conventions 13
- Online Documentation Errata 13

Chapter 1: Overview 15

- TeamSite Elements 15
 - Backing Stores 15
 - Branches 15
 - Workareas 16
 - Staging Areas 16
 - Editions 17
- TeamSite User Roles 18
 - Authors 18
 - Editors 18
 - Administrators 18
 - Masters 19
- TeamSite Workflow 19
 - Workflow Models 19
 - Jobs 20
 - Tasks 20
- TeamSite Architecture 21

Chapter 2: Installing TeamSite 23

- Before You Begin 23
 - Hardware Requirements 24
 - Software Requirements 26
 - TeamSite Client Requirements 27
 - TeamSite Default File Locations 28
- Basic Installation 29
 - Installing TeamSite 29
 - Obtaining a License Key 31
 - Installing the License Key 33
 - Rebooting Your System 35
 - Installing TeamSite Templating 35



Upgrading to TeamSite 5.5.1	35
Configuring Web Servers	40
Running the IIS Auto Configuration Script	40
Specifying the Web Server Port Number	41
Specifying the Web Server httpd User Name	41
Configuring the iw-mount Alias	41
Enabling Server-Side Include Requests	46
Stopping and Restarting the Web Server	51
Troubleshooting	51
Redirecting NSAPI HTTPS Requests	52
Setting Up TeamSite Clients	53
Using the Graphical User Interface	53
Using the File System Interface	55
Loading Content	56
Creating a Subbranch	56
Creating a Workarea	59
Populating an Initial Workarea	60
Submitting Files to the Staging Area	61
Publishing a New Edition	63
Uninstalling TeamSite	64

Chapter 3: Managing Access 69

Security	69
Users	70
TeamSite Roles Overview	70
Adding and Removing Users	72
Adding Users	72
Deleting Users	73
Access Control	74
Group Membership	75
Checking User Roles	76
Locking Models	77
Submit Locking	77
Write Locking	78
Permissions	78

Chapter 4: Configuring TeamSite Through the Interwoven Administration GUI 87

- About the Interwoven Administration GUI 87
 - Navigation 89
 - Apply, Refresh, and Cancel 90
- Logging In To the Interwoven Administration GUI 91
- Viewing System Information 91
- Editing Roles 93
- Setting Host Permissions 95
- Setting TeamSite Permissions 96
- Configuring General TeamSite GUI Preferences 98
- Changing Area Labels in the TeamSite GUI 100
- Configuring the General Proxy Settings 100
- Configuring Proxy Mappings 101
- Configuring Server Performance 103
- Configuring TeamSite Log Files 105
- Viewing TeamSite Log Files 107
- Performing Server Operations 109
 - Abort 109
 - Freeze or Unfreeze 109
 - Reset 110

Chapter 5: Configuring the TeamSite Server 111

- Configuring GUI Appearance 114
 - Configuring TeamSite Area Labels 114
 - Configuring Edition Views 116
 - Configuring History Views 117
 - User Profiles 117
- Configuring GUI Functionality 118
 - Disabling Editor Publish Capability 118
 - Enabling and Disabling SmartContext Editing 118
 - The Casual Contributor Interface: Adding Editing and Task Links to Web Pages 119
 - Setting the Default LaunchPad Interface 121
 - Setting Unique Server Names for LaunchPad to Recognize 121
 - Configuring Domain Lists in the Login Screen 122
 - Setting Login Authentication Expiration 122
 - Configuring Preview Windows 123
 - Custom Menu Items 125

Configuring Submit Button Behavior	131
Disabling Menu Items	132
Disabling Directory Operations	134
Disabling Unlocked File Auto-Upload	135
Setting the Number of Jobs Listed in the To Do List	135
Configuring Job Attribute Filters and Settings	136
Configuring Email Settings	137
Configuring Server Functionality	138
Specifying the Encoding of the iw.cfg File	138
User and Role Authentication Using LDAP	138
Using Domain Local Groups to Share Workareas	142
Webserver Group	142
Web Daemon	142
Servlet Engine	143
Main Branch Settings	143
Locked File Submission	144
Submit and Update Logs	144
Branch and Workarea Security	145
Domains to Use for Group Authentication	145
Logging Users and Groups	146
File Locations	146
Autoprivate	148
New File Templates	151
Launching Files Through iwproxy	153
Configuring the TeamSite Server Locale	154
Configuring Server Performance	155
Permitting Read-only Operations During Backing Store Freezes	155
Cache Size	155
RPCThreadcount	156
File System Threadcount	156
Filesystem Active Area Cache	156
Throughput Monitors	157
Detecting Low Disk Space	157
Submit Filtering	158
Configuring the TeamSite Web Daemon and Proxy Server	164
About the TeamSite Web Daemon	164
About the Proxy Server	164
Applying Changes to Proxy Configuration	166

Configuring TeamSite Web Daemon and Proxy Server Operation	166
Resolving Relative and Absolute Paths	167
Resolving Fully-Qualified URLs	171
Redirecting TeamSite Views to Different Areas	175
Configuring TeamSite to Use Different Web Servers	178
Configuring External Remappings	179
Host Header Remappings	180
Configuring SSI Remapping	181
Configuring Proxy Failover	181
Debugging Your Proxy Server Configuration	183
TeamSite Embedded Failsafe	184

Chapter 6: Configuring Metadata Capture and Search 187

Metadata Capture	187
Overview	188
Components	188
Configuring Metadata Capture	190
Metadata Capture End Result	210
Metadata Capture and TeamSite Workflow	211
Metadata Search	212
Overview	212
Prerequisites	212
Components	213
Configuring Metadata Search	215

Chapter 7: Managing the TeamSite Server 219

Checking Server Status	220
Verifying Server Operation	220
Checking Request Handling	220
Verifying the Server Mount	221
Finding the Installation Directory	221
Reviewing TeamSite Logs	222
Windows NT Event Viewer	222
TeamSite Log Files	223
Monitoring the Server Load	223
Starting and Stopping the Server	223
Managing the OpenAPI Server	224
Verifying that the OpenAPI Server is Running	224

Starting and Stopping OpenAPI	225
Reconfiguring iwwebd to Recognize a New IP Address	225
Re-Encrypting User Authentication Information	226
Troubleshooting	226
Troubleshooting TeamSite Access	226
Repairing the Backing Store	226
Managing Server Resources	232
Shared Directories and the TeamSite Mount Point	232
Changing TeamSite File Locations	232
Enhancing File System Performance on the TeamSite Server	232
Disk Space	232

Chapter 8: TeamSite Backing Stores 237

Backing Store Overview	237
Planning the Backing Store Conversion	238
Conversion Overview	239
Conversion Prerequisites and Tips	241
Converting Backing Stores Using the GUI	242
Converting Backing Stores from the Command Line	246
iwconvert.exe Command-Line Tool	246
Conversion Procedure	249
Creating Multiple Backing Stores	251
Defining Backing Stores in the iw.cfg File	252
Creating Backing Stores Using the iwstoreadm CLT	255
Administration CLTs	257
iwstoreadm.exe	257
iwidmap.exe	258
iwmigrate.exe	259
iwconvertserver.exe	260
SID Changes to the TeamSite Backing Store	260

Chapter 9: Backing Up TeamSite 263

Integrating with Third-Party Backup Solutions	263
Suggested Strategies for Incremental Backups	265

Appendix A: TeamSite Configuration Files 267

Location of iw.cfg	268
Location of Roles Files	268

Appendix B: Specifying Content Encoding 269

- regex_map Defined 270
 - Simple regex_map Example 271
- The regex_map Format 272
 - Rule Syntax 272
 - Regular Expression Syntax 273
 - Variables 273
 - Application Variables 274
 - Intermediate Variables 274
 - Interpolation of Variables and Captured Subexpressions 275
 - Quoting 278
- Strategies for Effective regex_maps 280
- Internationalization and regex_map 282
- SmartContext Editing and file_encoding.cfg 282
- Source Differencing and Merging and file_encoding.cfg 283
 - Sample file_encoding.cfg 284
- Advanced regex_map Example 285

Appendix C: High Availability TeamSite 287

- HA Watchdog 287
 - About HA Watchdog 287
 - TeamSite HA Watchdog Components and Processes 288
 - Installing TeamSite HA Watchdog 290
 - Configuring TeamSite HA Watchdog 290
 - Starting and Stopping the Server Under HA Watchdog 294
 - Uninstalling TeamSite HA Watchdog 294
 - Related Documentation 294
- HA Hot Standby 294
 - About HA Hot Standby 294
 - About Microsoft Cluster Server 296
 - Installing TeamSite and High Availability Hot Standby 296

Appendix D: Internationalization 305

- Supported Client and Server Platforms 305
 - Servers 306
 - Clients 306
 - Browsers 306
- Supported TeamSite Server Locales 307

Supported Content	307
Localization Overview	307
What's Been Translated?	307
What's Not Been Translated?	308
Limitations and Assumptions	309
Backing Stores and Character Encoding	310
About UTF-8	310
Interfacing with Localized Operating Systems	310
Accessing the Localized Interface	311
CLT Internationalization	311
CGI Internationalization	312
Specifying File Encoding of Text Files	313
Text Editor Encodings	314
Behavior of Netscape Navigator	314
Configuring Netscape for Multibyte Characters	315
Usage Scenarios	316

Appendix E: Client/Server Compatability 317

Index 323

About This Book

The *TeamSite Administration Guide* is a guide to installing, configuring, and maintaining TeamSite. It is primarily intended for TeamSite Administrators and Master users, and for web server administrators and system administrators.

Users should be familiar with either IIS or Netscape web servers, and with basic Windows NT operations such as adding users and modifying ACLs.

It is also very helpful to be familiar with regular expression syntax. If you are not familiar with regular expressions, it is recommended that you consult a reference manual such as *Mastering Regular Expressions*, by Jeffrey Friedl.

Some TeamSite configuration files make use of XML. For more information about XML, consult a reference manual or the online specification at <http://www.xml.com/axml/testaxml.htm>.

Notation Conventions

This manual uses the following notation conventions:

Convention	Definition and Usage
Bold	Text that appears in a GUI element (e.g., a menu item, button, or element of a dialog box) and command names are shown in bold. For example: Click Edit File in the Button Bar.
<i>Italic</i>	Book titles appear in italics. Terms are italicized the first time they are introduced. Important information may be italicized for emphasis.

Convention	Definition and Usage
Monospaced	<p>Commands, command-line output, and file names are in monospaced type. For example:</p> <p style="padding-left: 40px;">The <code>iwextattr</code> command-line tool allows you to set and look up extended attributes on a file.</p>
<i>Monospaced italic</i>	<p>Monospaced italics are used for command-line variables. The most common example of this is <i>iw-home</i>, which refers to the directory where TeamSite is installed. For example:</p> <p style="padding-left: 40px;"><i>iw-home\etc\iw.cfg</i></p> <p>is the path to the main TeamSite configuration file, <code>iw.cfg</code>, which is located in the <code>etc</code> directory under the TeamSite installation directory.</p> <p style="padding-left: 40px;"><code>iwckrole role user</code></p> <p>means that you must insert the values of <i>role</i> and <i>user</i> yourself.</p>
Monospaced bold	<p>Monospaced bold represents user input. The <code>></code> character that appears before a line of user input represents the command prompt and should not be typed. For example:</p> <p style="padding-left: 40px;">>iwextattr -s project=proj1 //IWSERVER/default/main/dev/WORKAREA/andre/products/index.html</p>
<i>Monospaced bold italic</i>	<p>Monospaced bold italic text is used to indicate a variable in user input. For example:</p> <p style="padding-left: 40px;">>iwextattr -s project=<i>projectname</i> <i>workareavpath</i></p> <p>means that you must insert the values of <i>projectname</i> and <i>workareavpath</i> when you enter this command.</p>
[]	Square brackets surrounding a command-line argument mean that the argument is optional.
	Vertical bars separating command-line arguments mean that only one of the arguments can be used.

Path Name Conventions

In most cases, you can specify path names using standard Windows NT naming conventions (which allow you to include spaces in path names). However, in some situations it might be necessary to use MS-DOS naming conventions, which stipulate that no single file or directory name in a path can contain a space or more than eight characters. If you encounter unexpected system behavior after

entering a path name using Windows NT naming conventions, enter the path name again using MS-DOS conventions. For example, instead of:

```
>C:\iw-home\Program Files\Interwoven
```

you can try:

```
>C:\iw-home\Progra~1\Interw~1
```

You can use the `dir /x` command to display the long and short versions of the file names in the current directory.

Online Documentation Errata

Additions and corrections to this document are available in PDF format at the following Web site: <http://support.interwoven.com>

When you reach this site:

1. Click **Download**.
2. Enter your user name and password.
3. Click **All Documentation**.
4. Click **Current Release Notes**.
5. Click the link to the appropriate PDF file.

Chapter 1

Overview

This chapter introduces the following three major TeamSite concepts and concludes with a description of the TeamSite system architecture:

- TeamSite Elements
- TeamSite User Roles
- TeamSite Workflow

TeamSite Elements

Backing Stores

The backing store is a large directory created by the TeamSite installation program that contains TeamSite files and metadata. By default, the backing store is located in `C:\iw-store`.

Previous releases of TeamSite have been limited to one backing store per TeamSite server. This release supports as many as eight backing stores per TeamSite server (the first created automatically by the installation program, and the others created by the TeamSite administrator). The functionality that enables multiple backing stores is known as *MultiStore*.

For detailed information about backing stores, MultiStore, and converting your existing backing store (upgrade customers only), refer to Chapter 8 and the *Backing Store Conversion Planning Guide*.

Branches

TeamSite provides *branches* for different paths of development for a Web site. Branches can be related to each other (for example, alternate language versions of the same Web site) or they may be completely independent. Each branch contains all the content for a Web site.



A single branch contains archived copies of the Web site as *editions*, a *staging area* for content integration, and individual *workareas* where users may develop content without disturbing one another. Branches can also contain *sub-branches*, so that teams may keep alternate paths of development separate from each other. Content can be easily shared and synchronized across branches and sub-branches. Users may work on one branch or on several, and the number of branches on a system is not limited.

Branches facilitate distributed workflow because they allow separate teams to work independently on different projects. Because all branches are located on the same TeamSite server, it is easy for one team to incorporate the work of another into their project.

Workareas

Each *workarea* contains a virtual copy of the entire Web site, which may be modified in any way without affecting the work of other contributors. Users who have access to a workarea may modify files within that workarea and view their changes within the context of the entire Web site before integrating their work with that of other contributors. Users can lock files in each workarea, eliminating the possibility of conflicting edits.

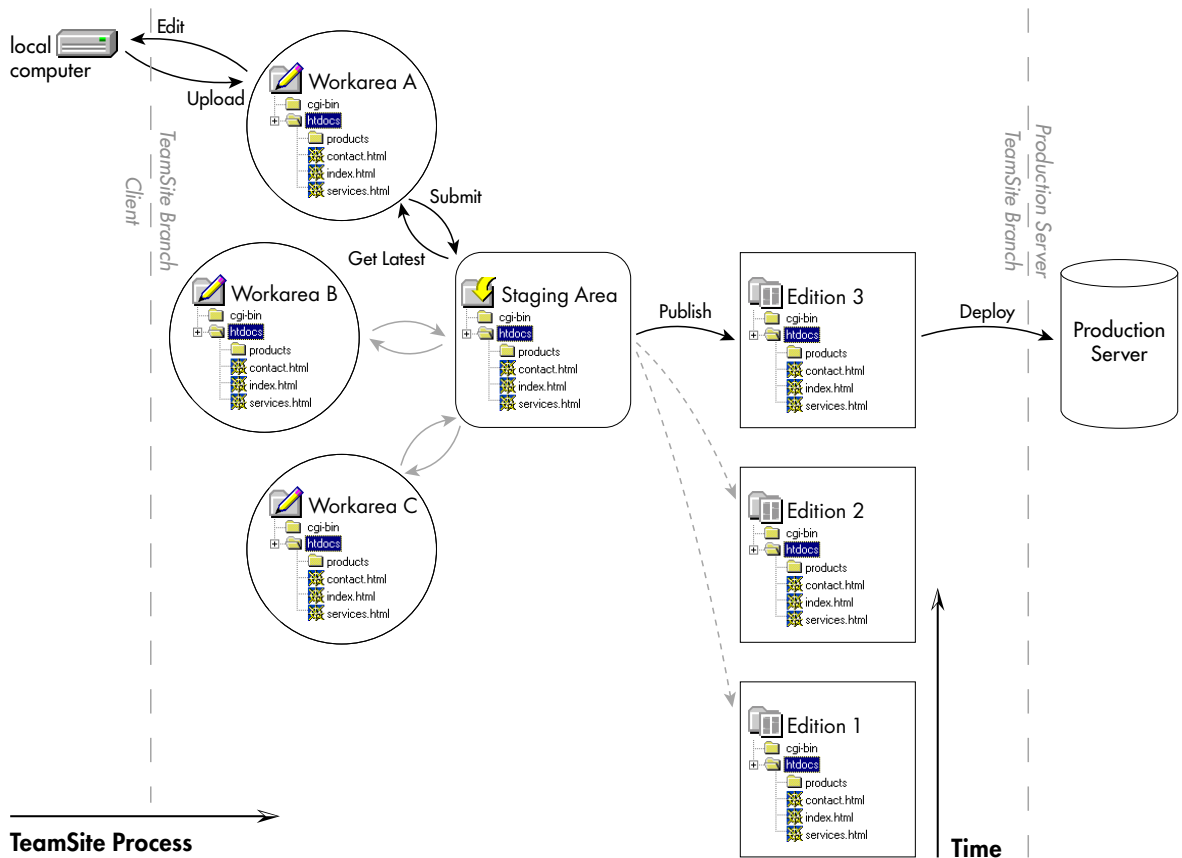
All changes that are made to files in a workarea are kept completely separate from other workareas and the staging area until the user chooses to promote his changes to the staging area. Within a workarea, users may add, edit, or delete files, or revert to older versions of files without affecting other users.

Staging Areas

Each branch contains one *staging area* where contributors incorporate their changes with the work of others. Users submit files from their workareas to the staging area to integrate their work with other contributions, and test the integrity of the resulting Web site. Because the staging area is an integrated component of the system, conflicts are easily identified and different versions of the same file can be merged, rather than overwritten.

Editions

Editions are read-only snapshots of the entire Web site, taken at sequential points in its development. Contributors can create new editions any time they feel their work is well integrated, or any time they want to create an update to the Web site for reference or deployment. Each edition is a fully functional version of the Web site, so that users may see the development of the Web site over time and compare it with current work.



TeamSite branches contain private workareas, which contain complete virtual copies of the Web site; staging areas, where contributors integrate their work; and editions, which are read-only snapshots of the Web site at various points in its development. Each area contains a virtual copy of the entire Web site. Content is submitted from workareas to the staging area, and the staging area is then published as an edition. Older editions are available for reference.

TeamSite User Roles

Authors

Authors are primary content creators. All work done by Authors goes through an explicit approval step. They can receive assignments from Editors, which are displayed in To-Do lists when Authors log in to TeamSite. Authors can access TeamSite from a simple browser-based interface, and do not need to be sophisticated computer users.

In order to test and QA their work, Authors have full access to the content in their Editors' workareas, but do not need to concern themselves with the larger structure and functionality of TeamSite. The Author role is appropriate for non-technical users, or for more technical contributors who do not need access to TeamSite's extended functionality, such as TeamSite's advanced version management features.

Editors

Editors own workareas. They create and edit content, just as Authors do, but they are primarily responsible for managing the development taking place within their workareas. This includes assigning files to Authors and submitting completed content to the staging area, and it may include publishing editions.

Editors have access to specialized TeamSite content and workflow management functions. Editors are generally "managerial" users, who primarily supervise the work of Authors, or self-managing "power" users, who need TeamSite's extended functionality to manage their own content.

Administrators

Administrators own branches. They have all the abilities of Editors, but they are primarily responsible for the content and functioning of their branch. Administrators can manage project workflow by creating new workareas for Editors and groups, and by creating sub-branches of their own branch to explore separate paths of development.

An Administrator is the supervisor of the project being developed on his branch. He may be the webmaster for a particular version of the Web site, or a project manager.

Masters

Master users own the Web site. They can perform all the functions of Editors and Administrators on any branch. The Master user owns the main branch, from which all sub-branches are created. The Master user is generally involved in the installation of TeamSite, and can reconfigure TeamSite on a system-wide basis.

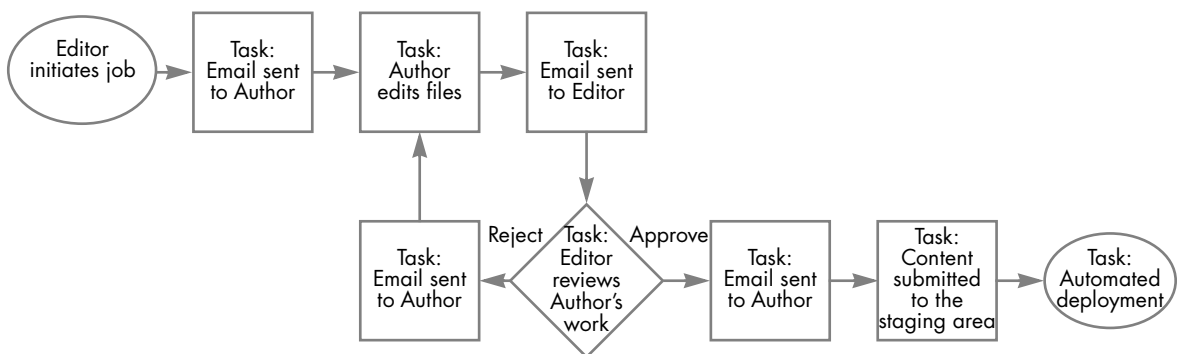
TeamSite Workflow

Workflow Models

A *workflow model* is a general workflow configuration that can be used repeatedly. Each workflow model describes a process which may include user tasks and a wide variety of automated tasks. Workflow models are configured by the system administrator or by the Interwoven Client Services organization.

For more information about configuring different workflow models, consult the *TeamSite Workflow Developer's Guide*.

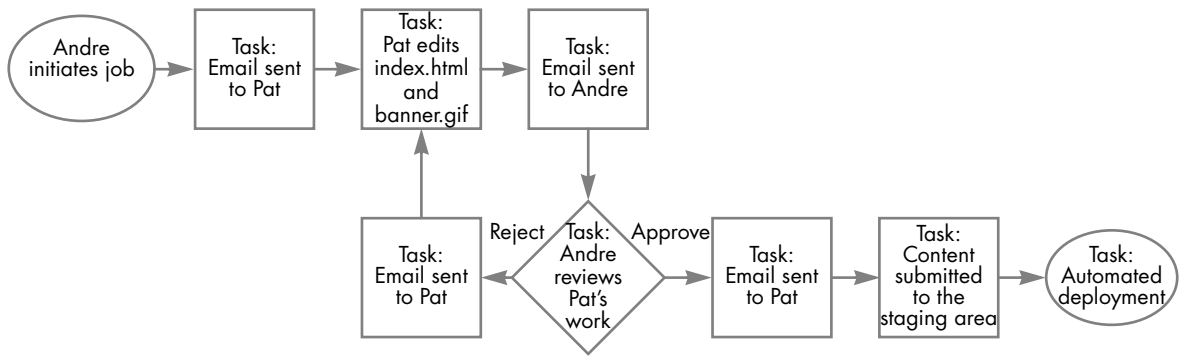
Below is a diagram of a very simple assign-edit-approve workflow model. Email is sent to the participants at every stage of the process, and some automated tasks are performed at the end.



Jobs

A *job* is a set of interdependent tasks. One example of a TeamSite job would be the set of tasks needed to prepare a new section in a marketing Web site to support a new product launch.

Each job is a specific instance of a workflow model. When a job is created, the job creator must supply all the specific information for that job. For example, the workflow model above might be used to create the job below.



Because jobs follow predefined workflow models, tasks cannot be added to or removed from individual jobs.

Tasks

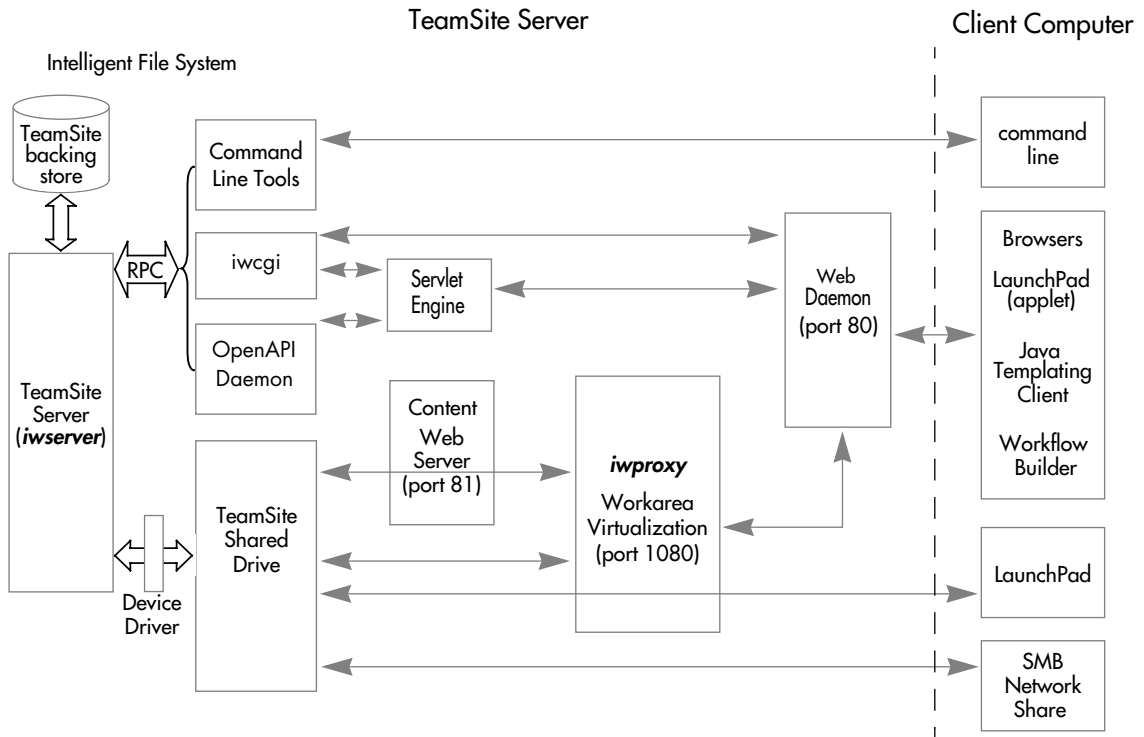
A *task* is a unit of work performed by a single user or process. Each task in a job is associated with a particular TeamSite workarea and carries a set of files with it. The user or process owning a task can modify, add files to, or remove files from the task.

Tasks have two possible states: active and inactive. A task becomes active when its predecessor tasks signal it to do so (predecessor tasks and conditions for activation are all configured as part of the workflow model). Once the task has been activated, users or external programs can work on it. For example, once a user task has been activated, the user can work on the files contained in the task. Once an external task has been activated, the appropriate external program can run on the files contained in the task. Inactive tasks are tasks that have been completed, or that have not been activated yet.

TeamSite Architecture

TeamSite's Intelligent File System (IFS) is composed of the TeamSite server and device driver, the TeamSite backing store of files and metadata, a suite of command-line tools, TeamSite CGI, proxy servers for access through the TeamSite browser-based GUI, and file system mounts for access through the file system interface.

The Intelligent File System is the core of the TeamSite system, where detailed information about the Web site, the web assets, web asset metadata, the production process and the users is stored. The Intelligent File System collects and maintains metadata on TeamSite files, directories, and areas, and allows TeamSite to process and present information according to who is asking for the information, and under what conditions. By using an object oriented design within a file system architecture, TeamSite combines extensive meta-data tagging with open access and file system performance for web content.



The client computer connects to the TeamSite server in several ways. Requests from the browsers or LaunchPad are routed through the TeamSiteWeb daemon, which allows consistent views of TeamSite areas. The double proxy server redirects hard-coded links within the Web site. Requests through the file system interface (TeamSite shared drive) and command-line tools, which do not go through the webserver, are not routed through a proxy server.

Chapter 2

Installing TeamSite

This chapter explains the process for installing TeamSite 5.5.1 and configuring all related system resources. The following topics are covered:

- Before You Begin—Describes TeamSite hardware, software, and client requirements.
- Basic Installation—Describes the basic steps for installing TeamSite 5.5.1.
- Upgrading to TeamSite 5.5.1—Describes the process for upgrading to TeamSite 5.5.1.
- Configuring Web Servers—Describes how to configure your Web server to work with TeamSite.
- Setting Up TeamSite Clients—Describes how to set up clients to access TeamSite.
- Loading Content—Describes how to transfer your existing Web site files into TeamSite.
- Uninstalling TeamSite—Describes how to uninstall TeamSite from your system.

Before You Begin

Before installing TeamSite, ensure that your system is configured with adequate hardware, software, and client resources as described in the following sections.

Hardware Requirements

This section addresses

- The amount of disk space your TeamSite system will need (page 24)
- How many CPUs you will need (page 24)
- Memory requirements (page 25)
- Recommended disk configuration (page 26)
- Requirements for the optional Global Report Center (page 26)

Disk Space

Your system must have at least 500 MB of disk space for the TeamSite program files, plus an additional five to 10 times the total amount of disk space you expect the Web site content files to consume.

Number of CPUs

The following recommendations are based on the concurrent numbers of different types of users who will be using TeamSite. This is because some types of users tend to do CPU-intensive operations such as Get Latest, Submit, or Compare, while other users tend to do more lightweight operations such as editing files and browsing directories.

All CPUs should be at least 700 MHz.

To determine the number of CPUs you need:

1. Determine how many of your users will be using TeamSite intensively (such as members of a Web development team), moderately, or mildly (such as occasional contributors to the company intranet).
2. Next, use these numbers to determine how many of each type of user will be using TeamSite concurrently. That is, those users who are actively interacting with TeamSite through one or more interfaces at the same time.

3. Locate the number of concurrent users in the chart below to determine the number of CPUs you need for each type of user, and add them together to get your total number of CPUs. You must have a minimum of two CPUs.

		Concurrent Users		
		Intense	Moderate	Mild
CPUs	1	25	50	100
	2	50	100	200
	4	100	200	400
	8 ¹	200	400	800

1. If the number of expected concurrent users is greater than the numbers in this row, please contact Interwoven Consulting Services.

For example, if you have a total of 60 intense, 250 moderate, and 2000 mild users, and you expect 40% of the intense users, 20% of the moderate users, and 10% of the mild users will be using TeamSite concurrently, then your concurrent users and total number of required CPUs would be as follows:

$$\begin{array}{rclcl}
 60 \text{ intense users} * .4 & = & 24 \text{ concurrent intense users} & = & 1 \text{ CPU} \\
 250 \text{ moderate users} * .2 & = & 50 \text{ concurrent moderate users} & = & 1 \text{ CPU} \\
 2000 \text{ mild users} * .1 & = & 200 \text{ concurrent mild users} & = & 2 \text{ CPUs} \\
 & & \text{Total} & & \underline{4 \text{ CPUs}}
 \end{array}$$

Memory

In general, TeamSite requires 1 GB of memory for each CPU (see above). To calculate memory requirements more precisely, use the following formula:

(1 GB for TeamSite and associated programs) + (cache size setting * 4 KB) + (4.6 MB * total number of concurrent users)

where the cache size setting is specified using the `cachesize` parameter in `iw.cfg`. By default, this setting is 30,000. For information on changing this setting, see “Cache Size” on page 155.

For example, the memory requirements for the example system specified above (which has 24 + 50 + 200 concurrent users), with the default cache size setting, would be:

$$1 \text{ GB} + (30000 * 4 \text{ KB}) + (4.6 \text{ MB} * (24 + 50 + 200)) = 2.38 \text{ GB}$$

If you encounter a significant amount of memory swapping, you should either increase the `cachesize` setting in `iw.cfg` or install more memory.

Disk Configuration

For maximum disk space efficiency, the TeamSite backing store should be installed on drives formatted with a 512 byte cluster size. For ease of maintenance, you may want to install the backing store on its own partition.

It is recommended that you use RAID 0+1 to configure your environment. RAID 5 can also be used for environments with a relatively low number of writes as a percentage of total accesses. Because TeamSite environments generally have a large percentage of writes, RAID 0+1 should provide better overall performance.

In addition to using RAID configurations, it is recommended that you use the fastest available SCSI controllers (160 MB/Sec transfer rate) and SCSI drives (10,000 RPM).

Note: Software RAID solutions are not recommended because they are very CPU-intensive.

Global Report Center Requirements

The TeamSite Global Report Center is an optional installation that requires an additional 25 MB of disk space, plus 10-50 MB for data storage. The Global Report Center also requires approximately 5 MB of physical memory. The OpenDeploy Global Report Center has the same requirement. Therefore, you should plan on approximately 10 MB of physical memory for the Global Report Center if you install both TeamSite and OpenDeploy.

Software Requirements

TeamSite runs on the same system as your Web site development server. It is recommended that you configure your Web site development server as a dedicated server. It should not run applications other than the Web server software and TeamSite.

The following software is required or recommended to run TeamSite on US operating systems (refer to Appendix D, “Internationalization” for details about other operating systems):

- Operating system—Windows 2000 SP2, Windows NT SP5 or greater
- Web server software—Microsoft Internet Information Server (IIS) 4.0 or 5.0, Netscape Enterprise Server 3.0, iPlanet 4.1

TeamSite Client Requirements

End users access TeamSite through browser-based thin-client technology. The only hardware requirements for client systems are that the RAM, CPU, local storage, and networking capability must be sufficient to operate a web browser and the editing applications of the user’s choice. TeamSite’s thin-client interface does not require you to install any other client software unless you will be editing files through the TeamSite GUI.

Not all TeamSite features are compatible with all browsers on all client platforms. The following table shows compatibility for Netscape and Internet Explorer:

	Netscape	Internet Explorer
Windows 95, 98, and NT	4.76	4.x-5.5 ²
Windows 2000	4.76	5.0-5.5 ²
Solaris (Sparc) 2.6, 7, and 8	4.76	Not supported
MacOS 8.6–9.x	4.76 ¹	5.0

1. Interwoven Merge not supported on Netscape for MacOS.
2. Some versions of Internet Explorer 5.5 do not include the Java Virtual Machine.
If you do not have the Java Virtual Machine you can download it from Microsoft's Web site at www.microsoft.com.

Note: If you are using Netscape browsers to display multi-byte characters, you must select **Edit > Preferences > Appearance > Fonts** and set the **Use my fonts, overriding page-specified fonts** option.

Connecting Through the File System Interface

To connect to TeamSite via the file system interface, users must have a network connection and the ability to connect to the TeamSite-shared IFS volume via their local domains. For more information, see “Using the File System Interface” on page 55.

TeamSite Default File Locations

By default, TeamSite is installed in the following locations (you may select alternate locations for some of these files during the installation process):

Default Directory	Contents
C:\Program Files\Interwoven\TeamSite	Default location of TeamSite program files. The location of this directory may be changed during installation or when the server is stopped. This directory is often installed in a location with a shorter path, such as C:\iw-home. With some system configurations, this shorter path is necessary to allow configuration of the iwperl and CGI areas. Wherever it is located, this directory is referred to throughout this manual as <i>iw-home</i> .
C:\iw-store	<p>Default location of the TeamSite backing store (TeamSite storage of files and metadata for workareas and editions). This directory can consume large amounts of disk space. The location of this directory may be changed during installation or when the server is stopped. To find where this directory is located, use the command-line tool <code>iwgetstore</code> (see <i>TeamSite Command-Line Tools</i>).</p> <p>Note: The contents of this directory should never be edited by hand in any way. Tampering with this directory can irreparably corrupt the data stored in TeamSite.</p>
Y:\	Default location of the TeamSite Intelligent File System volume. This directory is used to access Web site data when working directly from the server. The location of this directory can be changed; however, Web server virtual directories must be updated to reflect this.

Note: The TeamSite installation defaults to the system root drive. In the example above, C:\ is used.

Basic Installation

This section explains the basic TeamSite installation process. The following topics are covered:

- Installing TeamSite
- Obtaining a License Key
- Installing the License Key

Note: If you are upgrading to TeamSite 5.5.1 you need to first proceed to the “Upgrading to TeamSite 5.5.1” section on page 35.

The installation process is recorded in the `iwinstall.log` file in `C:\iw-home\install`.

Installing TeamSite

To install TeamSite on your Web site development server:

1. Log in as Administrator to the system where you want to install the TeamSite server.
2. Insert the TeamSite installation CD-ROM and browse to the top-level directory.
3. Double-click on `Teamsite.exe`.

The TeamSite installation files are automatically extracted. After the files are extracted, you are prompted for the location where you want to install the TeamSite program files and the backing store.

4. Accept the default installation location for the TeamSite program files (`C:\Program Files\Interwoven\TeamSite`) and for the backing store (`C:\iw-store`) or click **Browse** to select another location.

Note: If you specify an alternate location for the program files or the backing store, you must use ASCII characters. Also, for ease of maintenance, the backing store should be located on its own partition.

You are prompted to confirm whether you want to `iwproxyRestart.reg` to the Registry.

5. Click **Yes** to add the entry.

You are prompted whether you want to configure IIS for use with TeamSite.

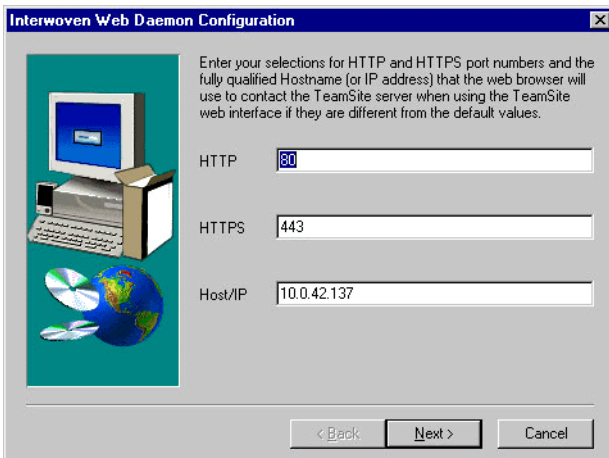
6. Click **Yes** to have the Setup program automatically configure IIS or **No** if you want to configure it manually later.

The Web Daemon Configuration window is displayed prompting you for port numbers for the Web daemon HTTP and HTTPS servers, and hostname or IP address.

Note: TeamSite now includes a Web daemon. By default, the HTTP server is assigned port 80. In past releases of TeamSite, port 80 was assigned to the Web server by default. If you do not use port 80 for the HTTP server, users must explicitly specify the alternate port number in the URL each time they access TeamSite.

7. Click **Next** to accept the default values, or edit the values and click **Next**.

The Web Daemon Configuration window is displayed prompting you for the Servlet, Proxy, and Web server port numbers.

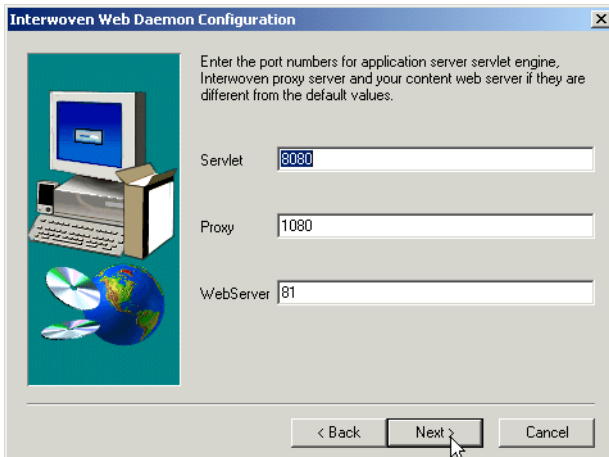


Specifying port numbers for the Interwoven Web Daemon server

8. For standard installations, accept the defaults by clicking **Next**.

The second Web Daemon Configuration window is displayed prompting you for the Servlet, Proxy, and Web server port numbers.

Note: The default port for the Web server (81) has changed from some previous releases of TeamSite. Ensure your Web server configuration reflects this change.



Specifying port numbers for Servlet, Proxy, and the Web server

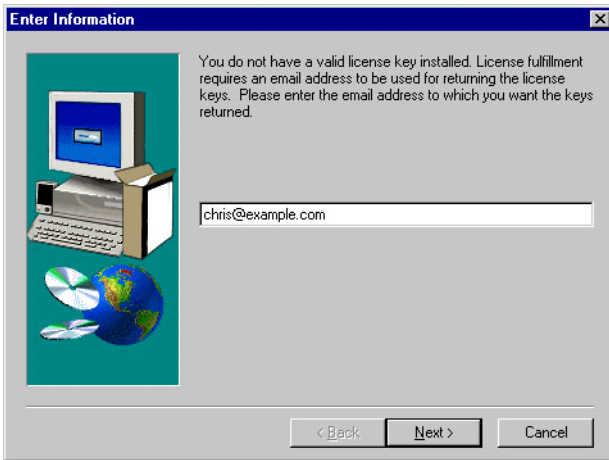
9. For standard installations, accept the defaults by clicking **Next**.

After your configuration setting are recorded, you are prompted for a TeamSite license key. Complete the procedure described in the next section to obtain a license key, or proceed to “Installing the License Key” on page 33 if you already have the license key.

Obtaining a License Key

Complete the following procedure to obtain a license key:

1. TeamSite will prompt you to enter an email address where you can receive a license key. Enter the appropriate email address, and click **OK**.



Entering an email address for the license key

2. Open the `tslicinfo.log` file in `iw-home/install` to obtain the information you will need to generate your license key.

```
hostname factotum
platform NT
domain example.com
hostMACid 80daac37
email iwautoprivat@example.com
product TS
version 5.5.1
os_version Windows NT 4.0
```

3. Log on to the Interwoven Support Web site's license generator page at:

<http://support.interwoven.com/license/license.asp>

or

<http://support2.interwoven.com/license/license.asp>

Follow the directions for obtaining a license key. After you enter the required fields, Interwoven will send a license key to your email address.

Installing the License Key

After you receive your license key from Interwoven support, you will need to install it. When TeamSite prompts you to enter a valid license key:

1. In the **Date** text field, enter the expiration date of your license. For example:
20010701
2. In the **License Key** text field, enter the appropriate license key. For example:
dd7629257a7b7c491a82641bb858c5a3 * host * 5.5.1 * admin@example.com

You can also enter your license key in the `iw.cfg` file. License keys have the following format:

```
license_expires=date
license_key=key * host * version * email_address
```

To enter your license key in the `iw.cfg` file:

1. Locate your `iw.cfg` file using the `iwgetlocation` command-line tool (located in `iw-home\bin`). For example, the following command:
>iwgetlocation -c iwconfig
The command in the example would return the following location:
C:\iw-home\etc\iw.cfg
2. Using a text editor, add the following lines to the `[iwserver]` section of the `iw.cfg` file as shown in the following example. You can copy and paste the `license_expires` and `license_key` lines from the license key email:

```
[iwserver]
license_expires=date
license_key=key * host * version * email_address
```

Some types of licenses may require additional lines to be added to the `iw.cfg` file. You will receive further information with your license key, if necessary.

After installation is complete, TeamSite will automatically execute the following post-installation procedures:

- Enable user impersonation, initially setting no password for the impersonation user

- Enable the SmartContext Editing (SCE) minimized tab
- Enable SmartContext QA
- Set up the Web Daemon
- Set up the Interwoven web application

Note: For security reasons, it is strongly recommended that you create a password for the impersonation user as soon as TeamSite is installed.

The installation procedure sets a Registry key (HKEY_LOCAL_MACHINE\Software\Interwoven\TeamSite) with the following values:

iw-home	The directory TeamSite was installed to (default: C:\Program Files\Interwoven\TeamSite)
iw-store	The directory for the TeamSite backing store (default: C:\iw-store)
Version	5.5.1

Troubleshooting License Keys

If your TeamSite server fails to run after you have installed your license key, look for diagnostic messages in the `iwserver.log` and `iwtrace.log` files.

Make sure that you installed your license key in the correct location by running the following command:

```
>iwgetlocation -c iwconfig
```

Use the command line utility `tsisvalid` (located in `iw-home\bin`) to verify that the license installed in `iw.cfg` is valid, as follows:

```
>tsisvalid iw-home
```

In the previous example, `iw-home` is the path to the TeamSite installation directory. The `tsisvalid` command creates a license status report file in `iw-home\install\tsisvalid.log`. If the license in the `iw.cfg` file is valid, `tsisvalid` prints the following line in the report file:

```
License is good.
```

If your license key is invalid, `tsisvalid` will print a report of possible reasons why it was not able to validate the license.

Rebooting Your System

After TeamSite is installed, you will need to reboot your system when prompted.

Installing TeamSite Templating

See the *TeamSite Templating Developer's Guide* for information about installing TeamSite Templating.

Upgrading to TeamSite 5.5.1

The TeamSite 5.5.1 upgrade procedure is the same as it has been for previous releases, but the circumstances under which you upgrade are dependant on the backing store conversion. In past releases you would load the new release onto the system where your current TeamSite server was installed. For TeamSite 5.5.1, the upgrade scenario is as follows:

1. With TeamSite 4.5.x or TeamSite 5.x installed on your current system (call it system A), install TeamSite 5.5.1 on a second system (system B).
2. Convert your old-format backing store (on system A) to the new high-performance backing store format on the system B. (This procedure is described in Chapter 8.)
3. If you want to use System A as your TeamSite 5.5.1 deployment server (as you have been), you must upgrade System A to TeamSite 5.5.1 and also migrate the new backing store (or stores) to system A.

Note: When copying the new backing store from one machine to another, you must ensure that all file attributes—including security and file times—are preserved.

The procedure for upgrading is described later in this section.

Note the following:

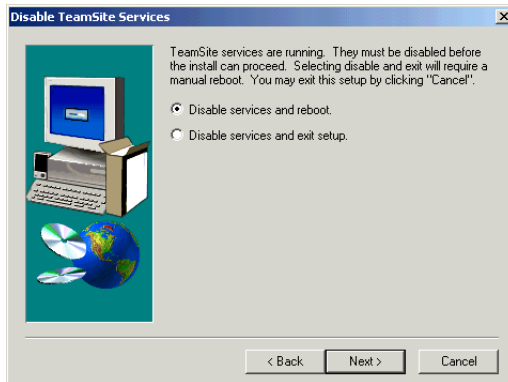
- The process of extracting TeamSite's installation files will overwrite all files contained in `iw-home` and its subdirectories, except for files that are not part of the default TeamSite installation.
- The contents of the `iw.cfg` file have changed for the 5.5.1 release. During the upgrade process, a copy of your existing `iw.cfg` file will be automatically renamed `iw.cfg.4.5.x` or `iw.cfg.5.0.x` and new `iw.cfg` and `iw.cfg.example` files will be created. If you had any customized settings in the previous version of your `iw.cfg` file that you want to apply to TeamSite 5.5.1, you will need to manually merge them into the new `iw.cfg` file.
- The default port numbers for the HTTP.
- If you are upgrading TeamSite Templating, the `available_templates.ipl` file is no longer used. It has been replaced by a new file called `available_templates.cfg`. If you modified the `available_templates.ipl` file and want to apply them to the 5.5.1 release, you will need to manually add your changes to the new `available_templates.cfg` file.
- If you do not have an existing TeamSite 4.5.x or 5.0.x installation on your system, you will need to install the appropriate TeamSite patch. For information on obtaining TeamSite patches, contact your Interwoven sales representative, or go to the following URL:

<http://support.interwoven.com>

To upgrade to TeamSite 5.5.1:

1. Back up your TeamSite `iw-store` directory, and your existing TeamSite configuration files, and roles files. These files include the following:
 - `iw-home\etc\iw.cfg`
 - `iw-home\local\config\templates.cfg`
 - `iw-home\local\config\autopivate.cfg`
 - `iw-home\conf\roles*.uid`
2. Log on to your system as Administrator.
3. Stop your Web server.
4. Insert the TeamSite installation CD-ROM and browse to the top-level directory.
5. Double-click the `TeamSite.exe` installation program.

When the installation program locates an existing TeamSite installation, it displays the Disable TeamSite Services window which prompts you to disable the active TeamSite services and reboot.

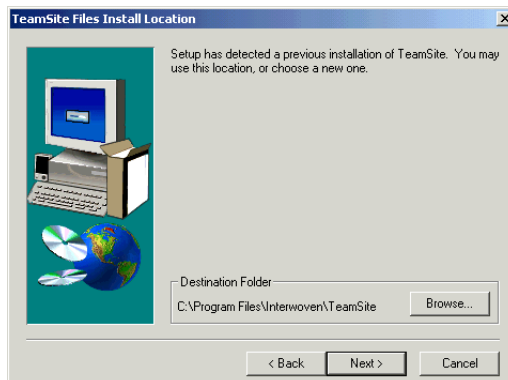


6. Click **Next** to disable the services and continue.

The Restarting Windows dialog box is displayed, prompting you to reboot your system now.

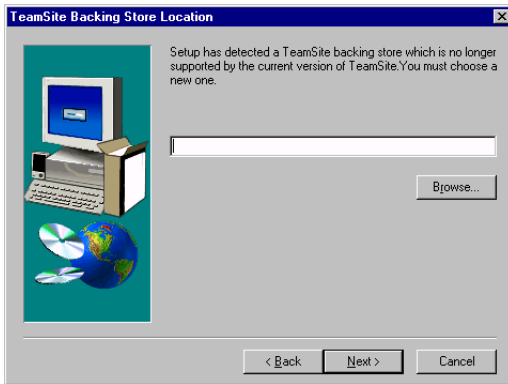
7. Click **OK** to accept the default and restart your system.
8. Log in to the system again as Administrator.
9. Double-click the TeamSite.exe to restart the installation program.

The TeamSite Files Install Location window is displayed.



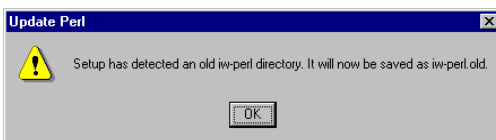
10. Click **Next** to accept the default installation location for the TeamSite program files (C:\Program Files\Interwoven\TeamSite) or click **Browse** to select another location.

The TeamSite Backing Store Location window is displayed, stating your old-format backing store has been detected and prompts you to select a different location for the new backing store.



11. Enter the location for the new backing store, for example: **C:\iwNewStore**.
12. Click **Next**.

A message is displayed stating that an old iw-perl directory has been detected and will be saved as iw-perl.old.



13. Click **OK** to continue.

The TeamSite program files are loaded into the location specified in step 10. You are then prompted to configure IIS for use with TeamSite.

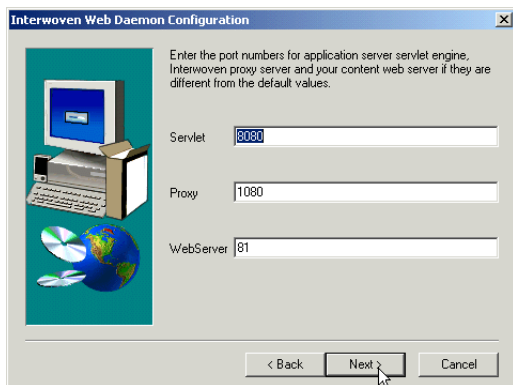
14. Click **Yes** to have the Setup program automatically configure IIS or **No** if you want to configure it manually later.

The Web Daemon Configuration window is displayed prompting you for port numbers for the Web daemon HTTP and HTTPS servers, and hostname or IP address.

By default, the HTTP server is assigned port 80. In past releases of TeamSite, port 80 was assigned to the Web server by default. If you do not use port 80 for the HTTP server, users must explicitly specify the alternate port number in the URL each time they access TeamSite.

15. Click **Next** to accept the default values, or edit the values and click **Next**.

The second Web Daemon Configuration window is displayed prompting you for the Servlet, Proxy, and Web server port numbers.



Note: The default port for the Web server (81) has changed from some previous releases of TeamSite. Ensure your Web server configuration reflects this change.

16. Click **Next** to accept the defaults.
17. After upgrading TeamSite, reboot the server.

Several Web server configuration changes have been made to the TeamSite 5.5.1 release. After the upgrade process is completed, proceed to the next section, “Configuring Web Servers.”

Configuring Web Servers

This section describes how to configure your system's Web server after you have installed or upgraded TeamSite. Configuration procedures for the following Web servers are discussed: Microsoft Internet Information Server (IIS), Netscape Enterprise Server, and iPlanet.

Note: It is important that you stop and restart your Web server after making any changes described in this section.

The following topics are discussed:

- Running the IIS Auto Configuration Script
- Specifying the Web Server Port Number
- Specifying the Web Server httpd User Name
- Configuring the iw-mount Alias
- Enabling Server-Side Includes
- Stopping and Restarting the Web Server
- Troubleshooting
- Redirecting NSAPI HTTPS Requests

Running the IIS Auto Configuration Script

During the TeamSite installation process, you were offered an option to automatically configure IIS. If you chose not to configure IIS during installation, you can use the automatic configuration script, `tspostreboot.pl`, at any time. The following example will automatically configure IIS to run with TeamSite:

```
>iw-home\iw-perl\bin\perl iw-home\install\tspostreboot.pl iw-home IIS
```

Note: This configuration script only applies to the IIS Web server. If you are using NES or iPlanet, you will need to manually configure your Web server.

Specifying the Web Server Port Number

It is important that the Web server port number specified in your Web server's `httpd.conf` file is consistent with the port number assigned in the `iw.cfg` file during the TeamSite installation. By default, TeamSite assigns port 81 to the Web server. If you are using TeamSite's default configuration, check to make sure that the port number in your `httpd.conf` file reads as follows:

```
# Port: The port to which the standalone server listens. For ports
# <1023, you will need httpd to be run as root initially.
#
Port 81
```

Specifying the Web Server httpd User Name

When TeamSite is installed, the user name `SYSTEM` is added to the `master.uid` file in `iw-home\conf\roles\`. If the `httpd` user name for your system is not `SYSTEM`, you must specify the correct user name in the `master.uid` file.

For information about how to edit the `master.uid` file, see “Adding and Removing Users” on page 72.

Configuring the iw-mount Alias

The `iw-mount` alias enables your Web server to access the default location of the TeamSite Intelligent File System volume. If you are upgrading to TeamSite 5.5.1, the `iw-mount` alias now incorporates the functionality of the former `iw`, `iw-bin`, and `iw-icons` aliases. The process for configuring `iw-mount` will vary depending on the type of Web server you are using.

If you are using Microsoft Internet Information Server (IIS), the IIS configuration script automatically creates and configures the properties for the `iw-mount` alias. It also creates a new web directory under the default website. If these tasks are completed, no further configuration of the `iw-mount` alias is necessary.

Since TeamSite release 5.0, the virtual directory `iw-mount` (which points to the `Y:\` drive) is created under a site called `Default Web Site` with a port number of 81. Currently, one of the following websites is chosen (in order of preference):

- `IW-Mount Web Site`—Selected if you reinstall TeamSite 5.5.1
- `TeamSite`—Selected if a previous TeamSite version is already installed

`Default Web Site`—IIS default

- Any website on the customer-selected port

If none of these are found, the installation script creates a website called `IW-Mount Web Site` on the customer-selected port, and naming it and placing it on the customer-selected port.

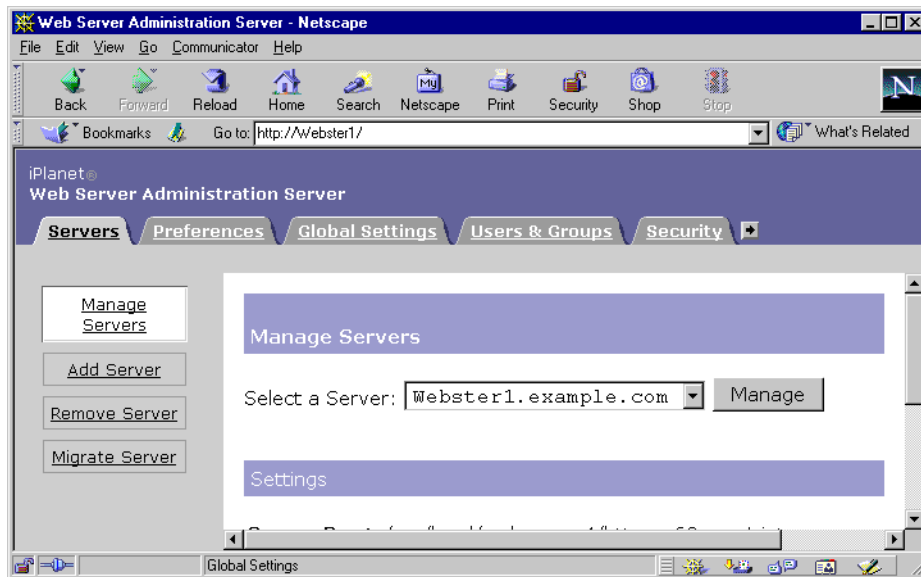
Configuring `iw-mount` for NES or iPlanet

You can configure the `iw-mount` alias for NES or iPlanet using the Server Administrator.

Note: For iPlanet, you can also configure `iw-mount` by editing the `obj.conf` file.

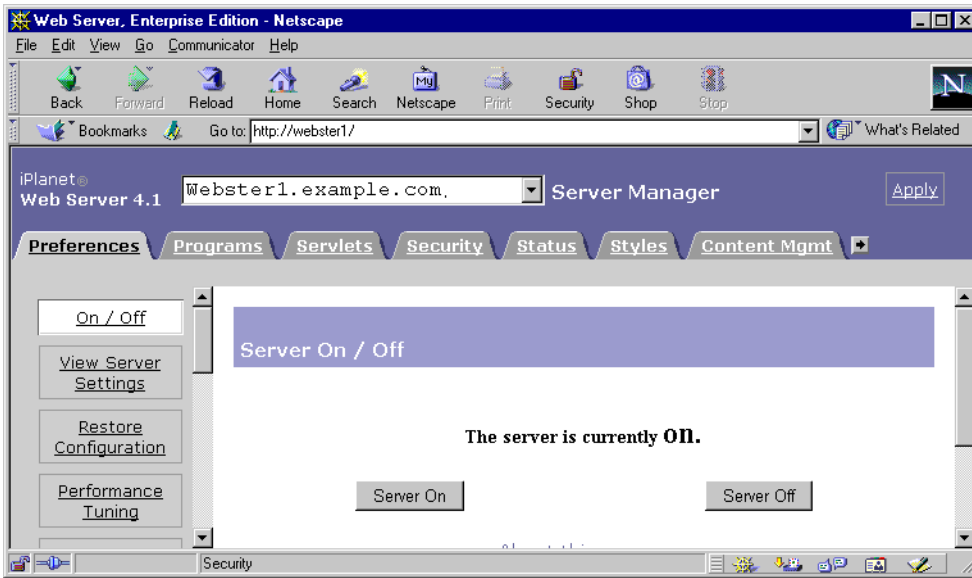
To configure `iw-mount` using the NES or iPlanet Server Administrator:

1. In the **Manage Servers** section of the Server Administrator, select the appropriate server name and click the **Manage** button.



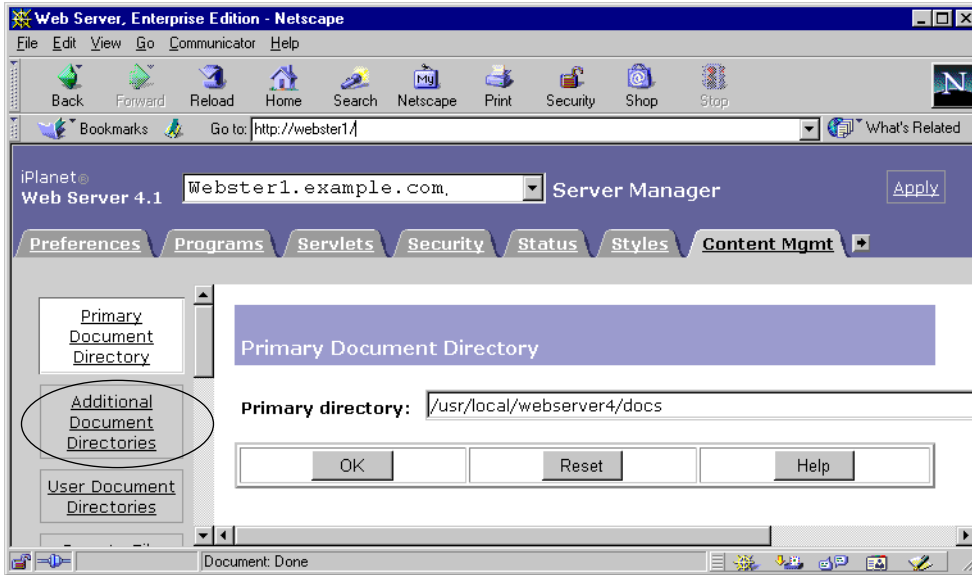
Selecting the server in the Manage Servers area of the Server Administrator

The Server Administrator will provide the status of the selected server.



Server status screen

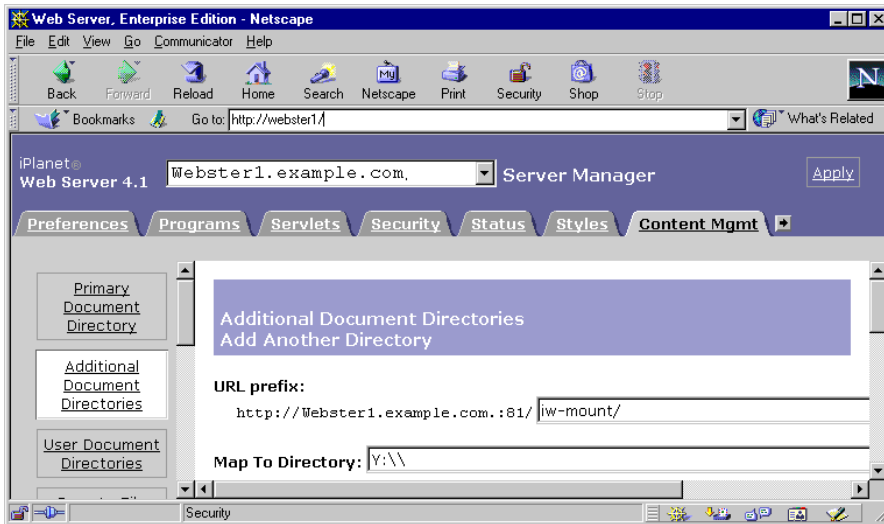
2. At the top of the Server Administrator, select the **Content Mgmt** tab.
The Content Management screen appears.
3. On the left panel of the Server Administrator, select **Additional Document Directories**.



Selecting Additional Document Directories

The Additional Document Directories screen appears.

4. In the Additional Document Directories screen, do the following:
 - In the **URL Prefix** text field, enter **iw-mount/**
 - In the **Map To Directory** text field, enter **Y:**



Specifying the iw-mount alias

5. Click **OK**, and save your changes when prompted.

For iPlanet, you can also configure the iw-mount alias by adding the following directive in the default object section of the `obj.conf` file:

```

NameTrans fn="pfx2dir" from="/iw-mount" dir="y:\\

```

Enabling Server-Side Include Requests

Because server-side include requests (SSIs) do not go through the proxy server, you must install TeamSite's redirector module to enable SmartContext QA for SSIs. The configuration steps will vary depending on the type of Web server you are using.

You will also need to set up your web server to use server-side includes. Specifically, you may need to turn on parsing of `.shtml` files. For more information on this process, consult the NCSA server-side include tutorial at:

<http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html>.

Note: If your Web site does not use SSIs, you do not need to install the redirector module.

Installing the Redirector Module for NES and iPlanet

To install the redirector module for NES and iPlanet:

1. Perform the following edits in the `obj.conf` file:

- a. In the `Init` section add the two `Init` directives shown below, substituting the pathname to the `iwproxy_nsapi.dll` file as appropriate for your installation.

Note: In this instance, NES uses back slashes as separators, while iPlanet uses forward slashes.

For NES:

First entry (all on one line):

```
Init fn="load-modules" shlib="iw-home\lib\iwproxy_nsapi.dll"  
funcs="iwrewrite"
```

For iPlanet:

```
Init fn="load-modules" shlib="iw-home/lib/iwproxy_nsapi.dll"  
funcs="iwrewrite"
```

Second entry (NES and iPlanet):

```
Init fn="iwrewrite"
```

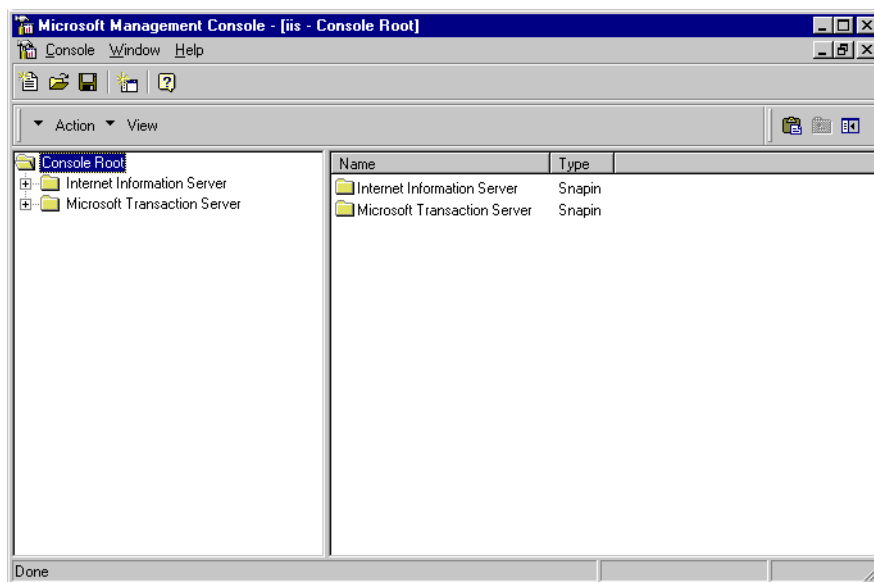
- b. In the default object description, add the following `NameTrans` as the first `NameTrans`, superseding all others:

```
NameTrans fn="iwrewrite"
```

Installing the Redirector Module for IIS

To install the redirector module for IIS:

1. Access the Microsoft Management Console.

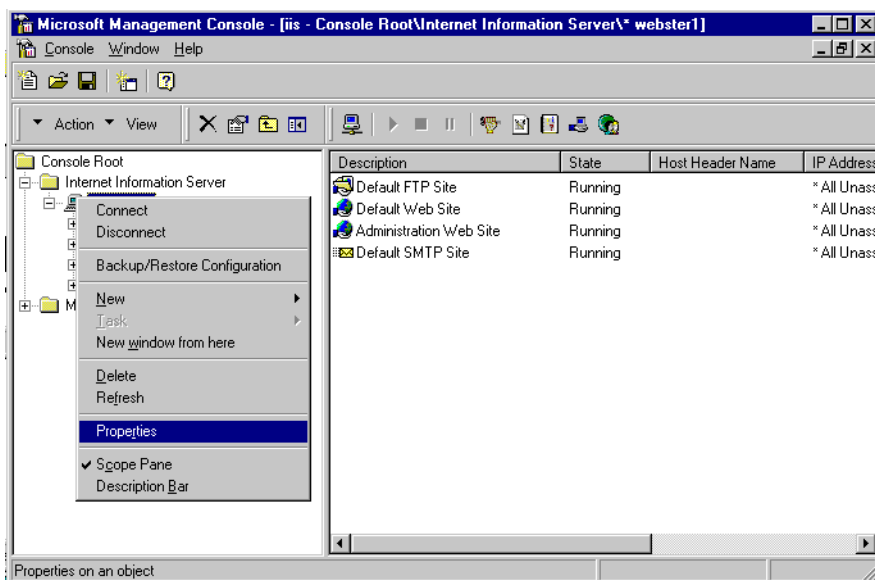


Microsoft Management console

2. Open the Internet Information Server folder.
3. Click on the icon associated with the name of your server.

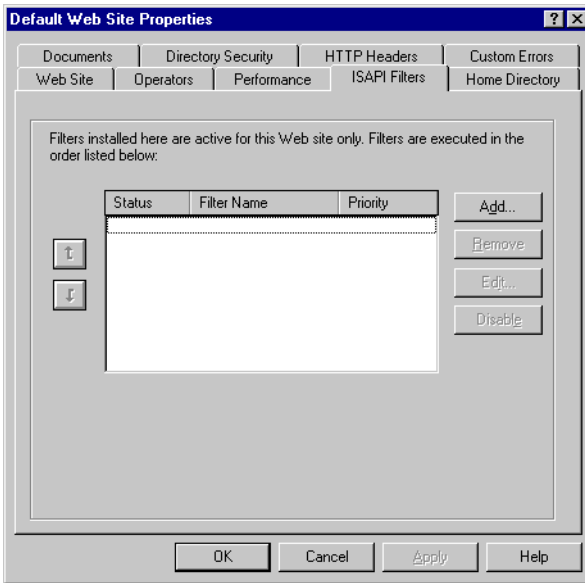
4. Right-click on the **Default Website** icon.

A drop-down menu appears.



5. In the drop-down menu, select **Properties**.

The Default Web Site Properties dialog box appears.

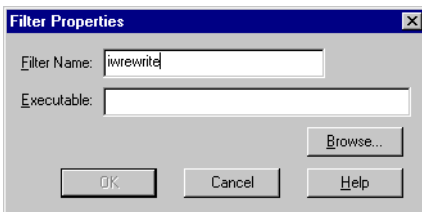


Default Web Site Properties dialog box.

6. In the Default Web Site Properties dialog box, click the **ISAPI Filters** tab, then click the **Add** button.

The Filter Properties dialog box appears.

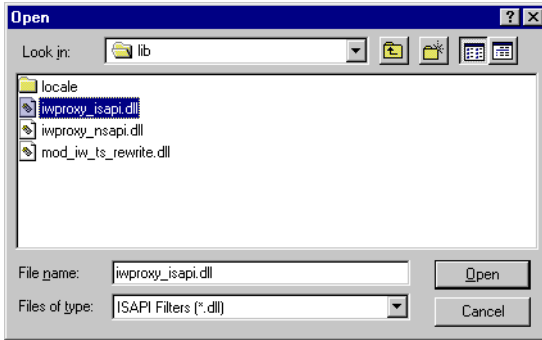
7. In the Filter Properties dialog box, enter **iwrewrite** in the **Filter Name** text field.



Filter Properties dialog box

8. Click the **Browse** button and browse to the **iwproxy_isapi.dll** file in the following directory:

Interwoven\TeamSite\lib



Selecting the iwproxy_isapi.dll file

9. Click **Open**.

10. Click **OK** in the Filter Properties dialog box.

TeamSite's redirector module for SSIs is now configured.

Stopping and Restarting the Web Server

Before you reboot your Web server, make sure that `_docroot` lines in the `[iwproxy_remap]` section of the `iw.cfg` file do not end with a trailing slash.

After ensuring that all `_docroot` trailing slashes in the `[iwproxy_remap]` section are deleted, you are ready to stop and start your Web server.

Troubleshooting

If you are using an IIS Web server and SSI requests are not executing properly, try the following procedure:

1. Open the Microsoft Management Console.
2. Right-click on the name of your Web server.
3. Select **Properties**.
4. Select the **Home Directory** tab.

5. Select **Configuration**.
6. Select the **App Mappings** tab.
7. Confirm that files with an `.htm` extension are mapped to the following file:
`WINNT\system32\inet_srv\ssinc.dll`
If the application mapping does not exist, create it.
8. Apply changes and confirm by selecting **OK**.

Redirecting NSAPI HTTPS Requests

You can configure TeamSite to redirect HTTPS requests from TeamSite so they are served from the Web daemon (`iwwwebd`) over HTTP. To do this, your system must contain two Web servers:

- A secure NES, IIS, or iPlanet Web server set up to process HTTPS requests.
- A non-secure server of any type that processes TeamSite web daemon HTTP requests.

To redirect HTTPS requests, set the following directive in the `[nsapi]` section of the `iw.cfg` file:

```
redirect_https_to_http=yes
```

When redirection is enabled, all HTTPS requests originating from the browser and received by the secure server's NSAPI plugin are redirected to the web daemon. The web daemon then sends the requests to the non-secure TeamSite server just as it would any request originating from the browser. For example, if the NSAPI plugin on the secure server receives an HTTPS request for a file in a TeamSite area such as:

```
https://teamsite_host/iw-mount/branch1/STAGING/bio.html
```

where `teamsite_host` specifies the host name (such as `www.example.com`), then the request is redirected to the web daemon as follows:

```
http://teamsite_host:iwwwebd_port/iw-mount/branch1/STAGING/bio.html
```

where `iwwwebd_port` specifies the port number.

During the redirection process, some browsers could display a message warning that the request is being sent to an insecure document. This is normal browser behavior. If you see such a message, click **OK** to proceed. Note that HTTPS requests redirected to the web daemon no longer have HTTPS security.

Setting Up TeamSite Clients

After installing TeamSite and configuring your Web server, you will need to set up at least one TeamSite client. You can use either the graphical user interface or the file system interface for client access.

Using the Graphical User Interface

TeamSite's graphical user interface can be accessed through a JavaScript-capable web browser. In order to log in, you must be a TeamSite user. If you have not yet added users to TeamSite or changed your own user status, you should do so now (see Chapter 3, "Managing Access"). If you do not add users or change your own user status, you are limited to the TeamSite Master role, which is the default role for the user Administrator.

For detailed information about browser configuration, consult the *TeamSite User's Guide*.

1. To access TeamSite from a client computer, start your web browser and enter the following URL:

`http://TeamSite_hostname/iw/`

In the example, *TeamSite_hostname* is the name of the TeamSite server (for example, `teamsite1.example.com`). You may want to bookmark this URL for future use.

2. The TeamSite login screen will appear. In the login screen, select your user type (Author, Editor, Administrator, or Master) using the pulldown menu.

If you are logging in as an Author, you can select the interface you want to use. Authors can use either WebDesk or WebDesk Pro. WebDesk is the updated interface for TeamSite, and it provides superior ease of use and an updated look and feel. WebDesk Pro is provided for users who are familiar with earlier versions of TeamSite, although these users are encouraged to switch to WebDesk.

To use WebDesk, check the **WebDesk** check box. To use WebDesk Pro, do not check the **WebDesk** check box.

3. Enter your user name, domain name (defaults to your local domain) and password and click **Login**.



TeamSite Login screen

Installing LaunchPad

Before you can edit files using TeamSite, you will need to install TeamSite's helper application, LaunchPad. If you are already using an older version of LaunchPad, you will need to reinstall it (LaunchPad will prompt you to upgrade, if necessary). To download LaunchPad:

1. Log in to TeamSite through the browser interface.
2. Select **Edit > LaunchPad Setup**. See the "Getting Started" chapter of the *TeamSite User's Guide* for more information on installing, configuring, and using LaunchPad.

Using the File System Interface

The file system interface allows you to manage your web content in TeamSite as a shared network volume. The file system interface is used primarily for file management functions such as moving and copying files, and it can also be used to edit files. It also allows the use of links checkers and scripts that need to be able to access and/or create files. In addition, most TeamSite operations can be performed from the command line (see the documentation on Command Line Tools).

Windows 95, 98, 2000, and NT Clients

- The first time you access TeamSite from Windows, you may need to access the TeamSite server as a shared volume. The following instructions show how to access TeamSite via Network Neighborhood.

To access TeamSite from Windows, use Network Neighborhood to locate the TeamSite server. You can navigate to any directory in TeamSite (for example, the top level of a branch or workarea, or a directory within a workarea), and create a shortcut to that directory.

You can also mount the TeamSite server as a networked drive. To do this:

1. In Windows Explorer, select **Map Network Drive** from the **Tools** menu.
2. Select the drive letter you want to map the TeamSite server to from the pull-down **Drive** menu.
3. Locate the TeamSite server in the **Shared Directories** list. Double-click on the TeamSite server.
4. Double-click on `IWServer`.

The TeamSite server will now be mounted as a networked drive.

Loading Content

When you install TeamSite, the main branch is automatically created. It contains a staging area and an empty initial edition. Before you start using TeamSite for production, you must transfer your current Web site files into TeamSite. To populate TeamSite with your content¹ you will need to perform the following steps (detailed directions for each step follow):

1. Create a subbranch for your web developers.
2. Create a TeamSite workarea on the subbranch.
3. Populate the newly created workarea with existing files.
4. Set permissions on the files, or configure a submit filter (see “Submit Filtering” on page 158).
5. Submit the workarea to the staging area.
6. Publish an edition from the staging area.

The newly published edition will then become the foundation of all subsequent work done in TeamSite.

Creating a Subbranch

Interwoven recommends that all development take place on subbranches. The main branch is not usually used for development for several reasons. First, it requires a user with Master privileges to administer. In addition, if you are using TeamSite to develop multiple Web sites, development of one Web site on the main branch and other Web sites on subbranches may create a false hierarchy of branches—the subbranch will not necessarily bear any relation to the parent branch.

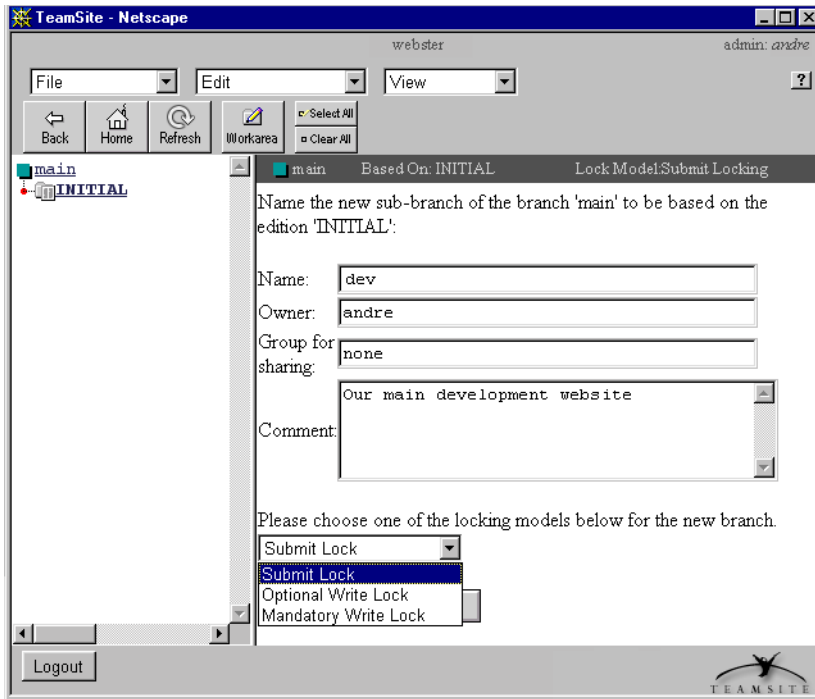
To create a subbranch using the TeamSite GUI:

1. Log in to TeamSite as a Master user.
2. Select **File > New Branch**. Because there is only one edition on the parent branch (the empty INITIAL edition), this subbranch will be based on that edition.

1. For issues regarding multibyte content, see Appendix D, “Internationalization”

3. A Create Branch window will appear.
4. Enter the name of the branch in the **Name** box. Do not use spaces or the following characters in the branch name:
 \ / : * ? " < > |
Do not name a branch WORKAREA, STAGING, or EDITION.
5. Your username will appear in the **Owner** box. If you want to assign the branch to someone else, type the owner's name in this box.
6. If you want this branch to have multiple Administrators, type the name of the group who will be able to administer this branch in the **Group for Sharing** box. The Administrator or Administrators of this branch will be able to create workareas and subbranches of development. For more information on Administrator privileges, see page 70 and page 78.
7. Use the pull-down menu to select the type of locking you want to be used on this branch (see the *TeamSite User's Guide* for an explanation of the different types of locking).
8. Add any comments in the **Comment** box (comments cannot be changed). Click **OK**.

Your newly created branch will contain no workareas, a staging area, and an empty edition called INITIAL.



Create Branch window

You can also use the `iwmkbr.exe` command-line tool to create a new branch (see *TeamSite Command-Line Tools*).

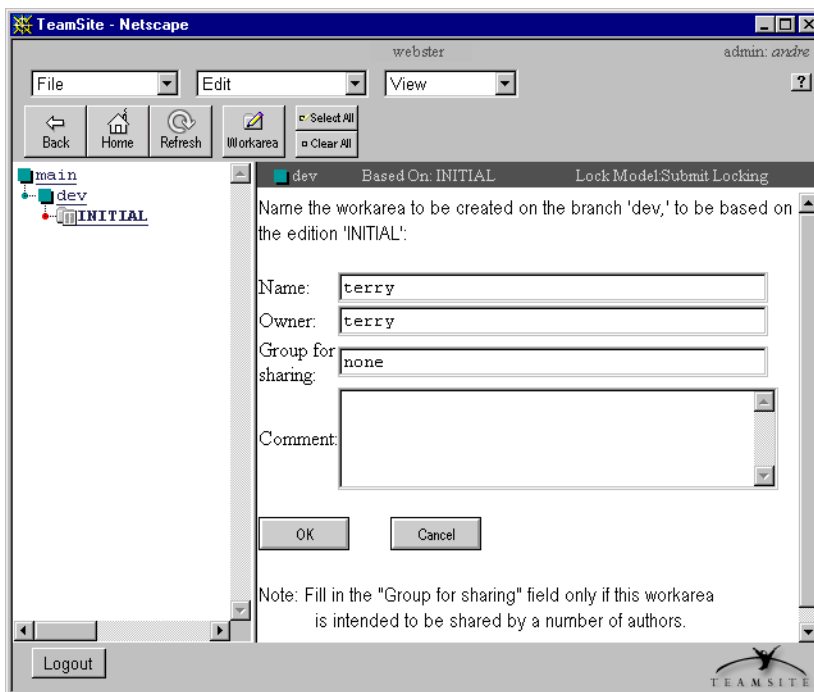
Creating a Workarea

To create a workarea using the TeamSite GUI:

1. Click the name of the subbranch you just created, to navigate into the branch.
2. Select **File > New Workarea**. Because there is only an empty edition on this branch, TeamSite will create an empty workarea.
3. The Create Workarea window will appear. Type in the name you want to give the workarea in the Name box, and the username (including domain) of the workarea's owner in the Owner box (for example, WEBSTER/andre).

Avoid using spaces and most punctuation characters in workarea names. Workarea names should consist only of alphanumeric characters, hyphens, and underscores.

4. Add any comments in the Comments box. Click **OK**.



CreateWorkarea window

You can also use the `iwmkwa.exe` command-line tool (see *TeamSite Command-Line Tools*).

Populating an Initial Workarea

To populate an initial workarea:

1. From the Windows NT or 2000 file system, log in as Administrator.
2. Copy all of the original Web site files into the new workarea (default location):

`Y:\default\main\branchname\WORKAREA\workareaname`

where *branchname* is the name of the newly created subbranch and *workareaname* is the name of the newly created workarea on the subbranch.

3. Set standard Windows ACLs on the files in your Web site.

Because TeamSite considers a change in ACLs to be a change in the file, TeamSite will store a new version of the file when you change its permissions (new versions are created at the time files are submitted to the staging area). If you wait to set permissions until after your files have been imported into a workarea and submitted to the staging area, you can create a large number of extra versions and unnecessarily clutter each file's version history. To avoid creating unnecessary versions, set permissions immediately after you populate the workarea (but before you submit the files). Interwoven recommends that you configure a submit filter to automate this process (see page 158), but you can also set permissions manually.

Note: Be sure to set permissions **before** you submit files to the staging area for the first time.

To set permissions on website files and directories, select either a file or directory, right-click and select **Properties** from the drop-down menu that appears. Click on the **Security** tab. In the Security window, add appropriate users and groups, and, in the case of a directory, determine whether to apply permission changes recursively down the entire directory tree.

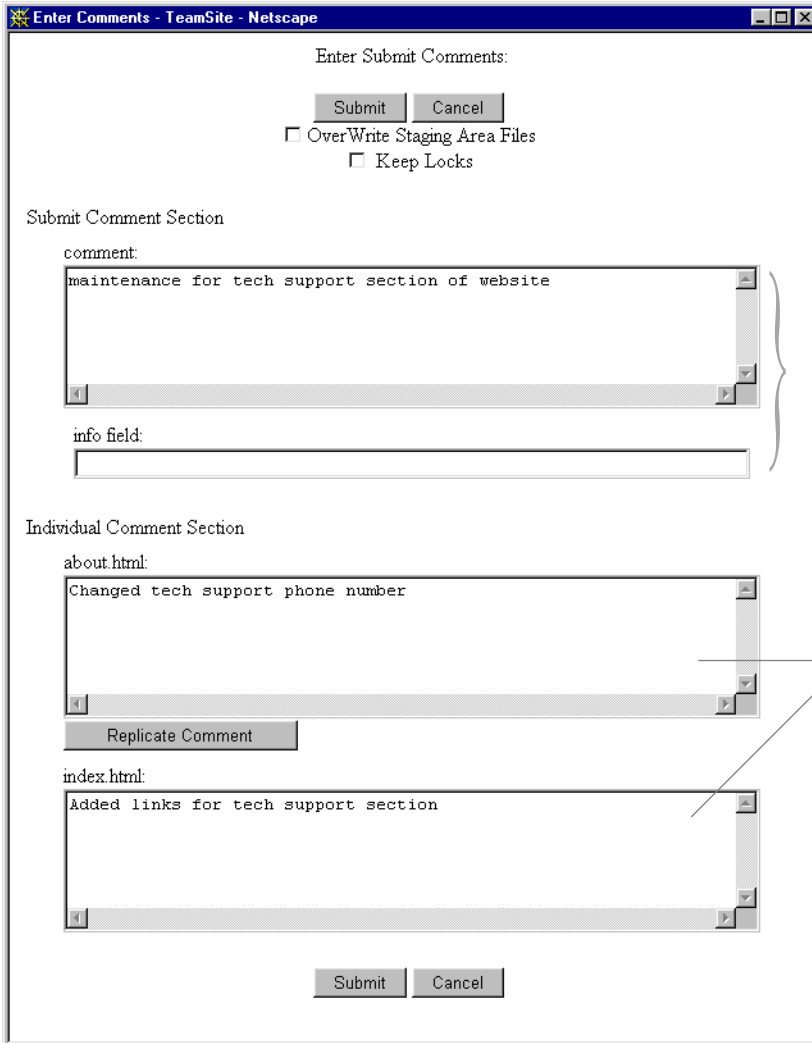
Submitting Files to the Staging Area

Now that you have populated your workarea with your Web site content, you can submit it to the staging area. You need to submit your content to the staging area before you can publish an edition, which you can use as the basis of all future workareas.

To submit the contents of your workarea to the staging area via the TeamSite GUI:

1. Go to the top level of the workarea. Do not select any checkboxes.
2. Select **File > Submit Direct**.¹ A dialog box will appear asking if you want to submit the entire directory.
3. Click **OK**. A Submit window will appear.
4. Enter any comments you have in the comment boxes. The Submit window contains two sections:
 - The **Submit Comments** section, which consists of a section where you can enter comments for the entire Submit operation, and a field where you can enter keywords, for example, for automatic triggers.
 - The **Individual Comments** section, where you can attach comments to each file.
5. Click the **Submit** button.

1. For first-time submissions of large numbers of files, you should use the direct Submit option rather than workflow Submit.



Submit Comments window

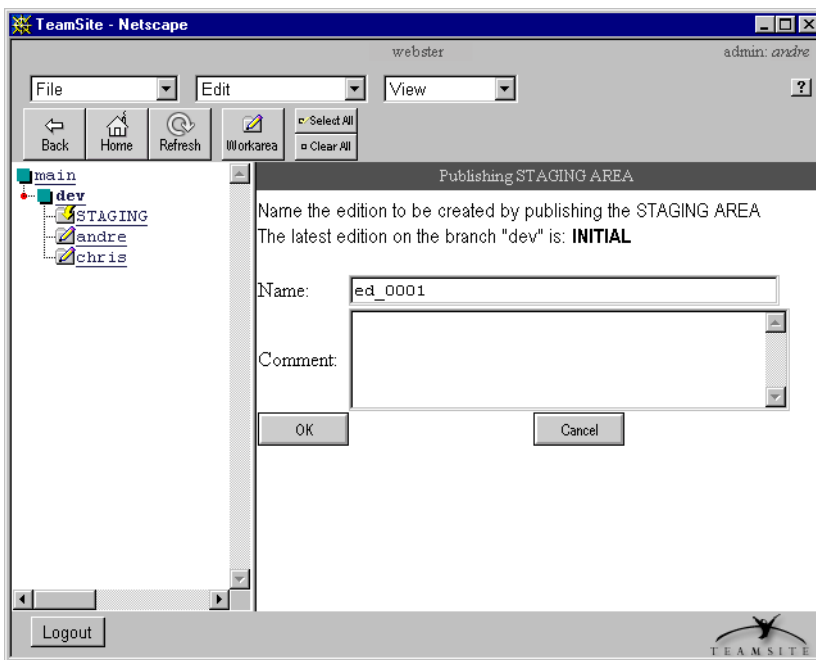
You can also use the `iwsubmit.exe` command-line tool to submit files to the staging area (see *TeamSite Command-Line Tools*).

Publishing a New Edition

Publishing an edition creates a snapshot of the staging area at the time of publication. These editions can be used as checkpoints. As part of your initial installation process, you should create an edition to record the state of your Web site at the time that you installed TeamSite. You can use this new edition as the basis for the other workareas you create on this branch.

To create a new edition from the contents of the staging area via the GUI:

1. Select **File > Publish** menu from anywhere within your branch to display the Publish window.
2. TeamSite will suggest a name for the new edition. If you want to give the edition a different name, enter the name of the new edition in the **Name** box in the Publish window.



Publish window

You can also use the `iwpublish.exe` command-line tool to publish an edition (see *TeamSite Command-Line Tools*).

Uninstalling TeamSite

If you are currently running TeamSite release 5.0.1 with Service Pack 1, you must uninstall the Service Pack before uninstalling TeamSite. To uninstall TeamSite 5.0.1 Service Pack 1:

1. Navigate to the directory that contains the NT_TS501_SP1.exe file.

The NT_TS501_SP1.exe file is the same file you used to install the service pack, you can locate it by using the operating system's Search functionality if you are not sure where you placed it during the installation.

2. Double-click the NT_TS501_SP1.exe.

When executed, the program checks to see if the service pack is installed, if it is, it starts the uninstall program.

3. Respond to the prompts to complete the uninstallation procedure.

Before you begin to uninstall TeamSite, you must ensure that you have access to the TeamSite.exe installation program that you used to originally install TeamSite. The installation program is used to stop certain services before you proceed with the TeamSite uninstall.

To uninstall TeamSite complete the following procedure:

1. Log in to the system where TeamSite is installed as Administrator.
2. Double-click the TeamSite.exe installation program.

When the installation program locates an existing TeamSite installation, it displays the Disable TeamSite Services window which prompts you to disable the active TeamSite services and reboot.

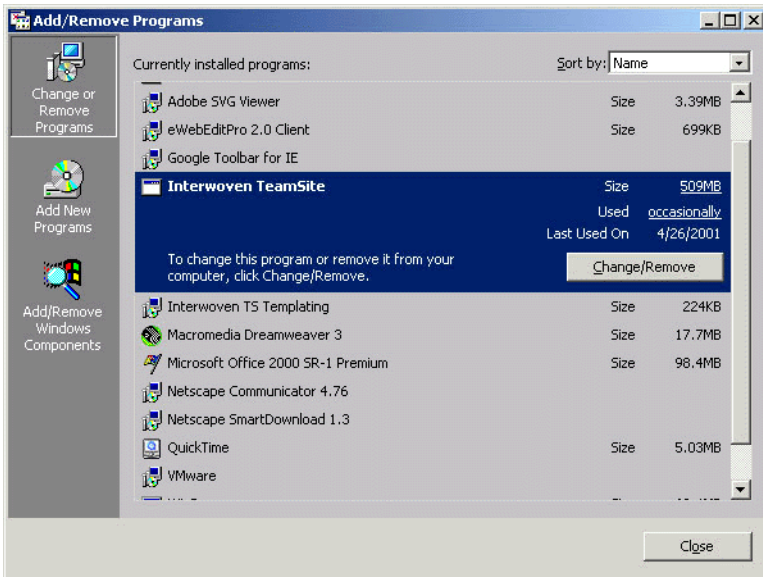


3. Click **Next** to disable the services and continue.

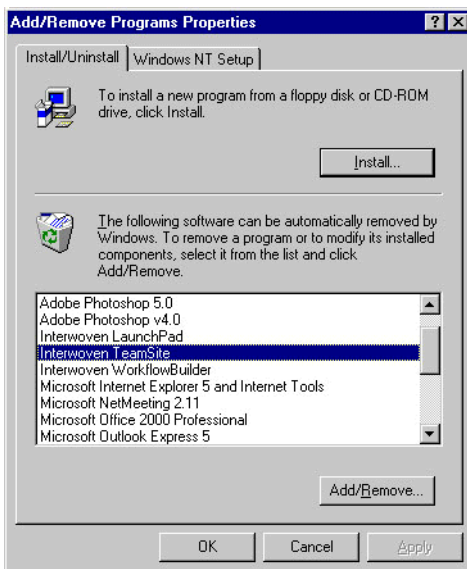
The Restarting Windows window is displayed prompting you to reboot your system now.

4. Click **OK** to accept the default and restart your system.
5. Log in to the system as Administrator.
6. Select **Start > Settings > Control Panel**.
7. Double-click the **Add/Remove Programs** icon.

The Add/Remove Programs window is displayed.



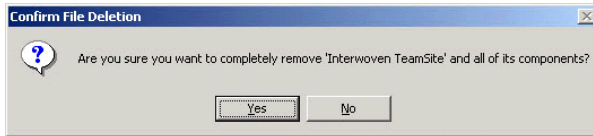
Add/Remove Programs control panel for Windows 2000



Add/Remove Programs Properties control panel for Windows NT

8. Select **Interwoven TeamSite** from the list of applications.
9. Click the **Change/Remove** button (for Windows 2000) or the **Add/Remove** button (for Windows NT).

The Confirm File Deletion window is displayed.



10. Click **Yes**.

TeamSite and its related files are removed from your system. If you plan to reinstall TeamSite, you must reboot your system again before doing so.

Chapter 3

Managing Access

Security

Access to TeamSite is governed by the following two factors:

- Windows-related permissions
- TeamSite access privileges

Windows file permissions control who has access to individual files and directories. Windows password authentication is used when logging in to TeamSite. However, TeamSite access privileges govern who can log in under various roles, and who has access to branches and workareas. For example, to edit a file in a workarea, a user must both be able to access that workarea (through TeamSite access privileges), and have permissions for that file and its parent directory (through Windows permissions). For a complete list of the TeamSite and Windows permissions needed to perform any action in TeamSite, see page 78.

When adding a new user, you need to take three factors into account:

- Whether the user has access to the server.
- The role the user will play in your Web site operations.
- The portion of the Web site the user will be editing.

If the user does not have access to the TeamSite server, you will need to add him (see page 72). To decide what TeamSite role best corresponds to the role he will play in your Web site operations, see page 70.

To decide what groups the new user needs to belong to, and which workareas he needs to access, consider your existing groups and which portions of the Web site and which workareas they can access. Add the new user to the groups that work on the same portion of the Web site that he will be editing, and he will automatically have access to their workareas and to their Web site files. If the new user needs his own workarea, create a private or shared workarea for him, but make sure that he owns or has access to the files that he will be editing. To change ownership or access to files, see page 74.

When creating a new workarea, you need to decide:

- What the name of the workarea should be.
- Who will need to access the workarea.
- What portions of the Web site the workarea's owner and group should and should not have access to.

Set permissions on your files according to the latter consideration. Remember that permissions cannot be set differently for different workareas. If the permissions for corresponding files are set differently in different workareas, you will encounter conflicts when you submit files to the staging area.

It is often useful to keep a chart of your Web site that shows what users and groups have access to what sections of the Web site.

Users

TeamSite Roles Overview

To facilitate workflow and security, TeamSite provides users with four roles, each with varying levels of access to TeamSite: Master, Administrator, Editor, and Author. To determine which role a user should have, consult the following table and find the role that best fits the user's work.

Author	Editor	Administrator	Master
Owens content	Owens workareas	Owens branches	Owens main branch

Author	Editor	Administrator	Master
Edits & creates files	Edits & creates files	Edits & creates files	Edits & creates files
Receives assignments	Assigns files	Assigns files	Assigns files
Work is approved by workarea owner	Approves/rejects work of Authors	Approves/rejects work of Authors	Approves/rejects work of Authors
	Uses advanced version management features	Uses advanced version management features	Uses advanced version management features
	Maintains content of workarea	Manages branch	Manages entire Web site
	Submits files to the staging area	Submits files to the staging area	Submits files to the staging area
	Publishes editions (optional)	Publishes editions	Publishes editions
		Creates and deletes workareas	Creates and deletes workareas
		Creates and deletes sub-branches	Creates and deletes sub-branches

In addition, Administrators can perform all the functions that Editors can. Master users can perform all the functions that Administrators and Editors can.

Adding and Removing Users

Adding Users

Before you can add a user to TeamSite, he must be a user on the TeamSite server or in a known domain. Always consult your system administrator before adding a user. If the user already exists, skip to Step 4.

To add a user to TeamSite:

1. From the **Start** menu, select **Programs > Administrative Tools > User Manager for Domains**.
2. In the User Manager, select **User > New User**. Fill in the **Username**, **Full Name**, **Description**, **Password**, and **Confirm Password** fields.
3. You can assign the password (in which case, make sure that the **Password Never Expires** checkbox is checked, and that none of the other checkboxes is checked) or allow the user to change it (in which case, make sure that the **User Must Change Password at Next Logon** checkbox is checked, and that none of the other checkboxes is checked).

Note: A user can only change his password when he is connecting via the file system interface. Therefore, the user must change his password before he first uses the TeamSite GUI. The new password will apply to all future connections through the TeamSite GUI.

4. Add the user's login name and domain name to the appropriate TeamSite roles file(s) (for example, MYDOMAIN\andre).

The four TeamSite roles files are in the directory *iw-home\conf\roles*. This directory may also contain roles files for other Interwoven products.

```
master.uid  
admin.uid  
editor.uid  
author.uid
```

Each file contains a list of the users who have privileges for that role, one user to a line.

To give a user multiple roles, include his name in multiple `.uid` files. For example, an Editor may also be able to log in as an Author. A Master user should be able to log in with any role, so you will need to include the name of each Master user in each `.uid` file. TeamSite users' passwords are the same as their Windows password.

After you have edited the TeamSite roles files, you will need to tell TeamSite to reread them. From the Command Prompt, type:

```
>iwreset
```

The TeamSite server will return 0 on success, non-zero on failure.

At this point the new user will now be able to log in to TeamSite, but he will not have access to any branches or workareas.

5. Add the user to the appropriate Windows groups. If you want the user to have access to a shared workarea, add him to the group that has access to that workarea. If the user is an Administrator, and you want him to have Administrator privileges for a branch, add him to the group of Administrators for that branch.

To add a user to a group, use the Windows NT User Manager. Locate the name of the group you want to add the user to, and add his name to the list.

6. If you want the user to own a workarea, create a workarea for him on the sub-branch where he will be working (see the *TeamSite User's Guide*). Be sure to include the user's domain name in the Name box of the New Workarea window (for example, WEBSTER\andre).

Deleting Users

To remove a user from TeamSite:

1. If your installation stores TeamSite role information in `.uid` files, delete the user's name from each of these files. If your installation uses LDAP to store role information, delete all the TeamSite roles from the user's LDAP entry.
2. Use `iwreset` to cause TeamSite to reread the user information from the roles files or LDAP database (see *TeamSite Command-Line Tools*).
3. Remove the user from TeamSite's entity database:

```
>iwuser -d DOMAIN\username
```

where `DOMAIN\username` is the username of the user you want to remove.

If you do not perform this step, you will not be able to create another user with the same name.

You might also want to remove the user's Windows user account for the TeamSite server. Use the Windows NT User Manager or, if Microsoft Active Directory is installed, use the Active Directory Users and Computers administrative tool:

1. Remove the user from any groups the user belongs to.
2. Remove the user account itself.

Access Control

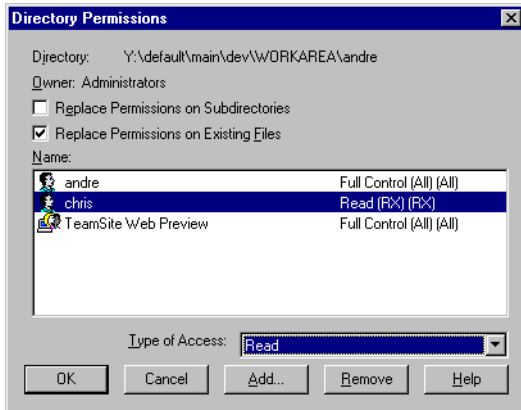
To control access to individual files or directories, use standard Windows NT permissions to change the file or directory's ACL. Newly created files will inherit permissions from their parent directory.

1. Navigate into the directory containing the file or directory (this example changes the ACL of a directory).
2. Select **Properties** from the **File** menu.
3. In the window that appears, click the **Security** tab. Click the **Permissions** button.



Setting permissions in Windows NT

4. A Directory Permissions window will appear, containing the directory's ACL. Use this window to add or remove users from the ACL, or to change the permissions of existing users.



A directory's ACL

You may also want to use a submit filter (see “Submit Filtering” on page 158) to automatically change and enforce permissions on files or directories.

Group Membership

Many workareas are shared by groups. For a user to have access to a particular workarea, he must either be the owner of the workarea, or a member of the workarea's group. TeamSite uses Windows groups for access control. These groups can be managed with the `iwchgrp` command-line tool. You can add as many users to a group as you want.

Changing Group Ownership of Workareas

To change which group has access to a workarea:

1. From the Command Prompt, navigate into the directory containing the workarea.
2. Use the `iwchgrp.exe` command (see *TeamSite Command-Line Tools*) to change the workarea's group.

Note: TeamSite only checks the primary group owner of a workarea, and does not rely on ACLs to determine workarea ownership.



Checking User Roles

The TeamSite Command Line Tool `iwckrole.exe` allows you to check whether or not a user can log in with a certain role.

Usage

```
iwckrole.exe [-h|-v] role user
```

<i>-h</i>	Displays usage message.
<i>-v</i>	Displays version.
<i>role</i>	author, editor, admin or master.
<i>user</i>	Username of the person whose role you are checking.

Exits with YES on successful authorization, NO on failure.

Example

```
>iwckrole admin andre
```

returns:

YES

indicating that user “andre” can log in as an Administrator.

Locking Models

TeamSite supports three different types of file locking: Submit Locking, Optional Write Locking, and Mandatory Write Locking. A branch's locking model is set when it is created (for more information, see the *TeamSite User's Guide*). Different branches on one TeamSite server may use different types of locking. All workareas on a branch use the same type of locking.

When a file is locked, it is locked for a particular workarea. That is, all users who have access to that workarea can edit the file. In addition, all users who have previously modified the file can edit it in their workareas (but not lock it).

Submit Locking

Submit locking means that if a file is locked, only users within the workarea where it is locked may submit the file to the staging area. Users are still allowed to edit the file within the context of other workareas but may not submit it until the user who holds the lock has submitted his version or manually released the lock.

If a file is not locked, anyone may submit it.

If someone else has the lock on a file that a user is editing, and the user tries to submit a workarea or directory containing his version after the lock holder has submitted the file and released the lock, the Compare Results window will appear showing the conflicting versions of the file. From this window, the user can choose to merge the two files, or to overwrite the version in the staging area with his own.

If someone else has locked a file, and a user edits it through the TeamSite GUI, the following warning is displayed:



The user can continue to edit the file, but will have to merge the changes with those of the lock owner after it is submitted.

Write Locking

Write locking means that a locked file may only be edited in the workarea where it is locked. Users in other workareas may not edit the same file even within the context of their own workareas, but they may view a read-only copy if they have the necessary permissions. Write locking may be optional, in which case users may choose whether or not to lock the files that they edit, or mandatory, in which case users cannot edit a file without locking it first. Under Mandatory Write Locking, all files in a workarea are read-only until a user locks them for editing. Once a user modifies a file while holding a lock, the user will be able to continue to modify the file even after releasing the lock. Once the user submits the file, it will become read-only again.

If a user edits an unlocked file, and somebody edits the file at the same time but in a different workarea, the second person to submit the file to the staging area will have a conflict and will need to either merge the two versions, overwrite the version in the staging area with his own, or overwrite his own version with the version in the staging area (see the *TeamSite User's Guide* for details).

Permissions

When a user tries to perform any action in TeamSite, the TeamSite server automatically checks to see whether or not he has permission to perform that action. TeamSite checks the following factors:

- User roles
- Branch permissions
- Workarea permissions
- File permissions
- Directory permissions

Not all of these factors apply to every action. TeamSite only checks the factors that apply to the action being attempted.

The table below lists which privileges a user must have in order to perform any action in TeamSite. To find out whether a user will be able to perform a specific action, check the entry for that action under the user's role and determine whether or not the specified conditions apply. All conditions

listed in each box below must apply in order for a user to perform an action, unless otherwise specified.

Note that TeamSite workflow tasks may require users to perform actions such as editing a file or submitting it to the staging area. To perform the task, the user must have the ability to perform the action as specified in the table below. For example, if you assign a task that requires an Author to edit a file, the Author must have workarea permissions, parent directory permissions, and file permissions for that file as specified in the table below.

User roles: If you are attempting to perform any of these actions through the GUI, you must be logged in with the role specified. If you are using the file system interface, you must be able to log in with the specified role.

Branch permissions: A user has branch permissions if he is either the primary owner of the branch or a parent branch, or if he belongs to the group that owns the branch or a parent branch. Master users automatically have branch permissions for all branches. Only Administrators and Master users can have branch permissions.

Workarea permissions: A user has workarea permissions if he is either the primary owner of a workarea, or if he belongs to the group that has access to the workarea. Workarea permissions are usually synonymous with Full Control permissions to the root directory of the workarea. If different permissions are specified, some sections of the table below will not apply.

File permissions: File permissions are Windows Change permissions (unless otherwise specified) to a file.

Directory permissions: Directory permissions are Windows Change permissions (unless otherwise specified) to a directory.

Task ownership: Includes the ability to take ownership of a task, for group tasks.

	Author	Editor	Administrator	Master
Edit file ¹	workarea permissions parent directory permissions (read/execute) file permissions (write)	workarea permissions parent directory permissions (read/execute) file permissions (write)	workarea permissions parent directory permissions (read/execute) file permissions (write)	workarea permissions parent directory permissions (read/execute) file permissions (write)
View file	workarea permissions parent directory permissions (read/execute) file permissions (read)	workarea permissions parent directory permissions (read/execute) file permissions (read)	workarea permissions parent directory permissions (read/execute) file permissions (read)	workarea permissions parent directory permissions (read/execute) file permissions (read)
New file ²	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)
Move file Rename file ³	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute) OR branch permissions	Yes
New directory	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)

	Author	Editor	Administrator	Master
Move directory Rename directory	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute) OR branch permissions	Yes
Delete file ⁴	No ⁵	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute) OR branch permissions	Yes
Delete directory	No	workarea permissions parent directory permissions (read/write/execute)	workarea permissions parent directory permissions (read/write/execute) OR branch permissions	Yes
Copy ⁶ (through the TeamSite GUI)	file permissions (read) directory permissions (destination directory) workarea permissions	file permissions (read) directory permissions (destination directory) workarea permissions	file permissions (read) directory permissions (destination directory) workarea permissions	file permissions (read) directory permissions (destination directory) workarea permissions

	Author	Editor	Administrator	Master
Lock file ⁷	No ⁸	workarea permissions parent directory permissions (read/execute) file permissions (write or ownership)	workarea permissions parent directory permissions (read/execute) file permissions (write or ownership)	workarea permissions parent directory permissions (read/execute) file permissions (write or ownership)
Unlock file	No	creator of lock OR owner of workarea	creator of lock OR owner of workarea OR branch permissions	Yes
Revert	No	workarea permissions	workarea permissions OR branch permissions	Yes
Get Latest	No	workarea permissions	workarea permissions OR branch permissions	Yes
Copy To	No	workarea permissions (destination workarea)	workarea permissions (destination workarea) OR branch permissions (destination)	Yes
Set Public/Private	No	workarea permissions	workarea permissions OR branch permissions	Yes

	Author	Editor	Administrator	Master
View History	No	workarea permissions (read)	workarea permissions (read)	Yes
Compare	No	workarea permissions (read)	workarea permissions (read)	Yes
List Modified	No	workarea permissions (read)	workarea permissions (read)	Yes
List Locks	No	workarea permissions (read)	workarea permissions (read)	Yes
View Submit Log	No	workarea permissions (read)	workarea permissions (read)	Yes
View Update Log	No	workarea permissions (read)	workarea permissions (read)	Yes
Create branch	No	No	branch permissions	Yes
Delete branch	No	No	branch permissions	Yes
Rename branch	No	No	branch permissions	Yes
Submit files	No ⁹	workarea permissions	workarea permissions OR branch permissions	Yes
Publish edition	No	workarea permissions for any workarea on the branch	workarea permissions for any workarea on the branch OR branch permissions	Yes
Delete edition	No	No	branch permissions	Yes
Rename edition	No	No	branch permissions	Yes
Create workarea	No	No	branch permissions	Yes

	Author	Editor	Administrator	Master
Delete workarea	No	No	branch permissions	Yes
Rename workarea	No	No	branch permissions	Yes
View reports	No	No	Yes	Yes
Assign file	No	workarea permissions	workarea OR branch permissions	Yes
View task	task ownership	task ownership OR job ownership	task ownership OR job ownership	task ownership OR job ownership
View job information	ownership of a task within the job	job ownership ownership of a task within the job	job ownership ownership of a task within the job	job ownership ownership of a task within the job
Create new job	No	available_templates.cfg setup	available_templates.cfg setup	available_templates.cfg setup
Transition task	job or task ownership	job or task ownership	job or task ownership	job or task ownership
Add/remove files from existing task	job or task ownership	job or task ownership	job or task ownership	job or task ownership
View task changes	job or task ownership	job or task ownership	job or task ownership	job or task ownership
Compare task files (with staging area)	job or task ownership	job or task ownership	job or task ownership	job or task ownership
Revert task files (to staging area version)	job or task ownership	job or task ownership	job or task ownership	job or task ownership

1. The ability to edit a file only applies to files that are not already write-locked. If the file is write-locked, then only the lock owner can edit it. Note that if an Author edits a file through the GUI, TeamSite will lock the file.
2. The ability to create a file only applies to files that are not already write-locked. You cannot create a file with the same name as a file that is already write-locked. If an Author creates a file, the new file will be assigned to him.
3. The ability to rename or move a file only applies to files that are not already write-locked. If the file is write-locked, then only the lock owner can rename it. A file cannot be renamed with the name of a file that is locked. If an Author renames or moves a file, the renamed or moved version of the file will be assigned to him.

4. For Authors and Editors, the ability to delete a file only applies to files that are not already write-locked. If the file is write-locked, then only the lock owner can delete it.
5. Authors can delete files through WebDesk only.
6. If an Author copies a file, the copied version of the file will be assigned to him.
7. The ability to lock a file only applies to files that are not already locked.
8. Authors can lock files through WebDesk only.
9. Authors cannot submit files directly. All work done by Authors must go through an approval process prior to submission. The approver must have the ability to submit files.

Configuring TeamSite Through the Interwoven Administration GUI

About the Interwoven Administration GUI

TeamSite includes a graphical user interface (GUI) framework that enables users to perform administrative tasks across a variety of Interwoven products. The Interwoven administration GUI is a Web application accessible from any system with a compatible browser (Netscape Navigator or Internet Explorer)

This chapter is limited to the TeamSite portion of the Interwoven administration GUI. For information on managing the settings for other Interwoven products through the Interwoven administration GUI, consult the documentation for those products.

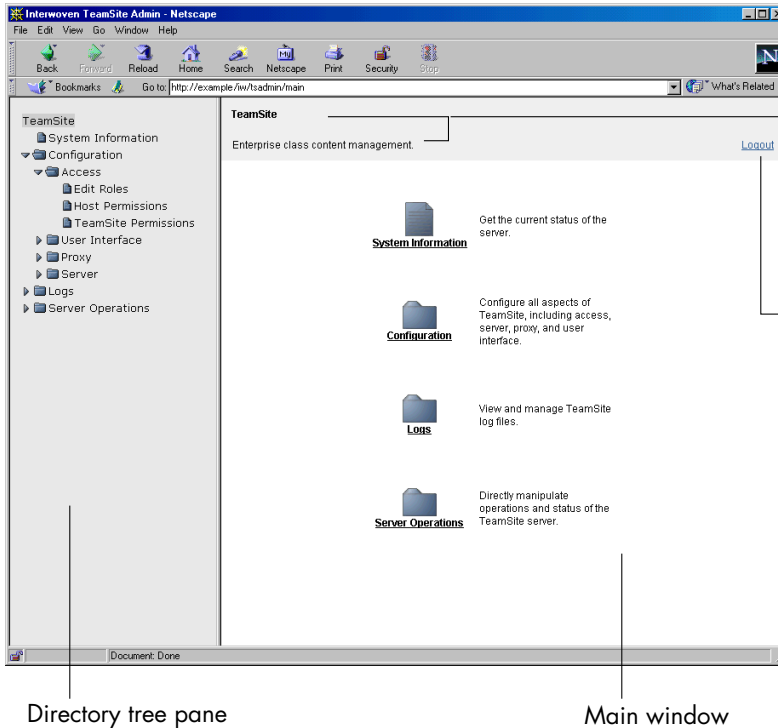
The TeamSite section of the Interwoven administration GUI provides an easy-to-use interface to:

- **View system information**—View basic information such as:
 - Server status
 - Version of the TeamSite server
 - License expiration date
 - Disk space
 - Load and throughput data
 - Server operations and the users who performed them

- **Configure the TeamSite client interface, proxy, and server**—Configure TeamSite without having to manually edit the main TeamSite configuration file, `iw.cfg`. When you apply a change to a setting through the GUI, the corresponding section in `iw.cfg` is edited. The TeamSite server automatically executes the changes when it next polls `iw.cfg` (usually within one minute).

When you apply changes through the GUI, the order of the options in `iw.cfg` and any comments therein are preserved. Additionally, a warning is displayed when you attempt to change a setting if any part of the `iw.cfg` file has been modified (either manually or through the GUI) between the time you loaded the current page and when you attempt to make the change. See “Apply, Refresh, and Cancel” on page 90 for details on responses to the warning.

- **View and configure log files**—View and configure these TeamSite log files:
 - `iwserver.log`
 - `iwevent.log`
 - `iwtrace.log`
- **Perform server operations**—Perform these operations on the TeamSite server without having to manually invoke command line tools (CLTs):
 - Abort
 - Freeze
 - Reset



Window title with navigation trail beneath (In the initial TeamSite window, a tagline replaces the navigation trail)

Logs you out of the GUI and returns you to the Login screen

Directory tree pane

Main window

The initial window of the TeamSite section of the Interwoven Administration GUI

Users who want to configure TeamSite by directly editing configuration files can find instructions for doing so throughout this book. Where appropriate, this chapter cross-references the more detailed information in other chapters regarding some configuration items.

Navigation

There are three ways to access the settings you want to change:

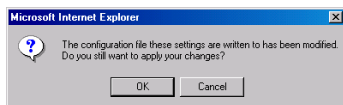
- Use the directory tree in the left-hand pane.
- Navigate through the folders in the main window.
- Use the links in the navigation trail under the window title.

Apply, Refresh, and Cancel

Most of the GUI windows have **Apply**, **Refresh**, and **Cancel** buttons near the bottom.

- **Apply**—Becomes active only when you change a setting. Click **Apply** to write your changes to `iw.cfg`.

If any part of the `iw.cfg` file has been modified between the last time you loaded your current page and when you click **Apply**, a warning dialog box is displayed.



Multiuser Warning

The warning is displayed whether or not the modified section is the same as the section you attempt to change. For example, user A logs in to the GUI and navigates to the Edit Roles. User A begins to enter data to add a user to TeamSite. At the same time, user B (either through the GUI or manually) changes a setting in the proxy section of `iw.cfg`. User A clicks **Apply**. The warning dialog box is displayed even though user A's change will modify a section of `iw.cfg` different than the section changed by user B.

Respond to the warning by doing one of the following:

- Click **OK** to apply your changes.
- Click **Cancel** to abort the process and exit the dialog box.
- Click the **Refresh** button in the GUI to abort the process and reload the page to display the most current settings.
- **Refresh**—Reloads the window so that the most current settings are displayed. In some cases (Edit Roles, for example) clicking **Refresh** clears all input fields.
- **Cancel**—Loads the enclosing title page of the window from which you canceled (for example, if you navigate to the Edit Roles window and click **Cancel**, you are taken to the Access page.) Changes are discarded.

Logging In To the Interwoven Administration GUI

You must have Master permissions on TeamSite to log in to the Interwoven Administration GUI.

To log in to the GUI:

1. Access the Administration GUI by typing the following in your browser:

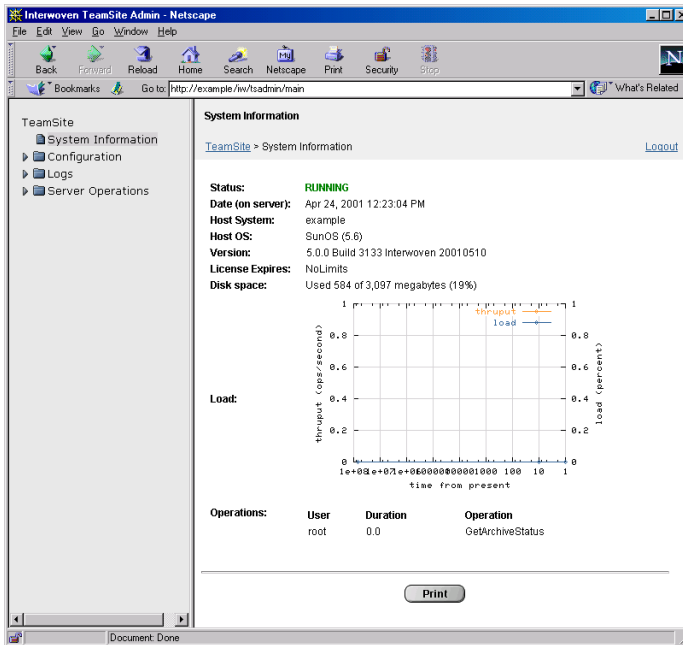
`http://hostname/iw/tsadmin`

2. Enter your TeamSite user name.
3. Enter your TeamSite password.
4. Click **Login**.

Viewing System Information

In the System Information window you can view:

- **Status**—Displays one of the following states:
 - Running—Any backing store is up or frozen.
 - Stopped—TeamSite is installed, but nothing is running.
 - Unknown—`tsadmin` could not determine the status.
- **Date (on server)**—Displays the date and time (as set on the server).
- **Host System**—Displays the name of the host system.
- **Host OS**—Displays the operating system of the host system.
- **Version**—Displays the version and build number of the TeamSite server.
- **License Expires**—Displays the license expiration date. For details about licenses, see page 33.
- **Disk space**—Displays the amount of disk space used on the host system.
- **Load**—Graphs average load and throughput against minutes of uptime. Data is supplied by the `iwstat` CLT.
- **Operations**—Lists the name, user, and duration (in seconds) of all active server operations.



System Information window

The System Information window is refreshed every 60 seconds.

To print system information, click **Print**.

Editing Roles

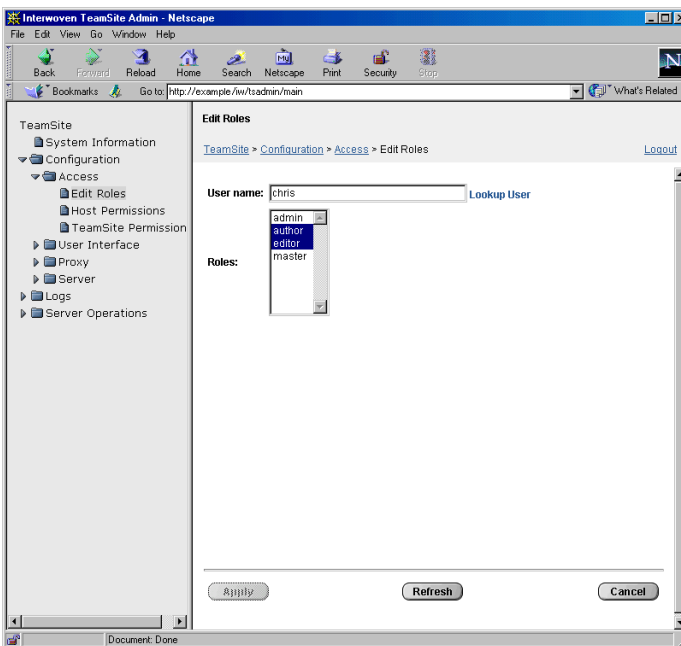
In the Edit Roles window you can:

- Add and remove TeamSite users.
- Edit the roles of existing TeamSite users.

The administration GUI ensures that the user name you enter represents a valid host system user, then displays a list of the TeamSite roles that the user can have.

To add a user to TeamSite, or to edit an existing TeamSite user's roles:

1. Navigate to **TeamSite > Configuration > Access > Edit Roles**.



Edit Roles window, displaying a sample TeamSite user

2. Enter the user name and click **Lookup User**.

Note: If the user is not on the host system, a message is displayed indicating this fact. You must add the user to the host system before you can add the user to TeamSite. To add the user to TeamSite, first add the user to the host system (consult your system's manual), then repeat steps 1-4 in this section.

3. Select the roles you want to give the user. (Hold down the Shift key on your keyboard to make continuous multiple selections. Hold down the Ctrl key on your keyboard to make discontinuous multiple selections.)
4. Click **Apply**.

The user is added to TeamSite with the specified roles.

To remove a user from TeamSite:

1. Enter the user name and click **Lookup User**.
2. Deselect all roles. (Hold down the Ctrl key on your keyboard to deselect the user's last remaining role.)
3. Click **Apply**.

A prompt is displayed asking you to confirm the removal of the user from TeamSite.

4. Click **OK** to remove the user from TeamSite.

Note: **Note:** If you inadvertently remove yourself from the Master role, you are immediately logged out of the GUI and can log back in only when another Master adds you back to the Master role.

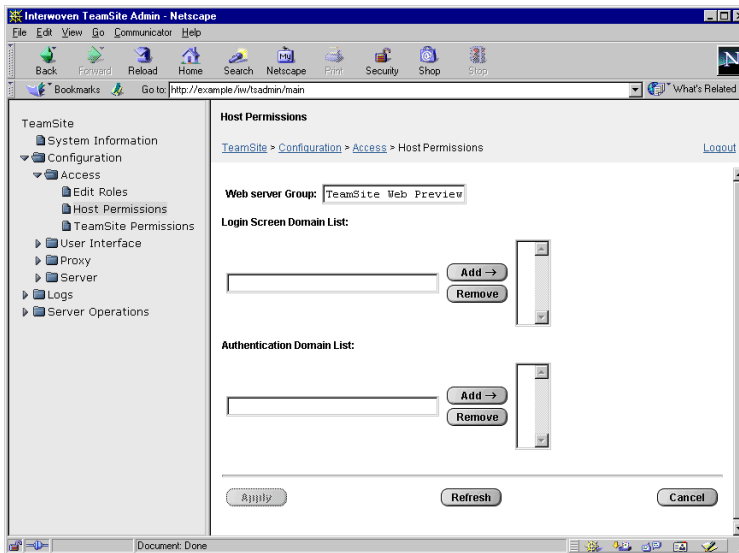
To add or remove a TeamSite user manually, see page 72.

Setting Host Permissions

In the Host Permissions window you can:

- Specify the Web server group.
- Edit the contents of the **Domain** pull-down menu in the TeamSite login screen.
- Edit the list of domains in the [iwserver] section of iw.cfg.

To configure host permissions, navigate to **TeamSite > Configuration > Access > Host Permissions**.



Host Permissions window

- **Web server Group**—Specifies the Web server group. You must enter the group of the Web server that communicates with TeamSite. Entering a different group here does not change the Web server's group.

A Web server runs as a particular user, usually *nobody*. In order for browsers to view Web content, TeamSite needs this setting to match the group of the Web server. If you do not enter a Web server group, "Everyone" is used by default.

- **Login Screen Domain List**—Lists the domains that display in the **Domain** pull-down menu of the TeamSite login screen. Any number of domains can be added to this list; however, you can reduce the time it takes the login screen to load by limiting the number of domains in this list. To configure this list manually, see "Configuring Domain Lists in the Login Screen" on page 122.
- **Authentication Domain List**—Edits the [iwserver] section of iw.cfg. This section lists domains that have group information necessary to TeamSite. Typically, the TeamSite administrator lists only the main company Primary Domain Controllers (PDCs) in the [iwserver] section of iw.cfg. Any number of domains can be added to this list; however, you can reduce the time it takes TeamSite to start up or to perform a user update—the time it takes TeamSite to recognize a new Windows user—by limiting the number of domains it tries to authenticate against.

Note: If a domain cannot be queried for group information, TeamSite must wait for a time out, typically 10-20 seconds per TeamSite user.

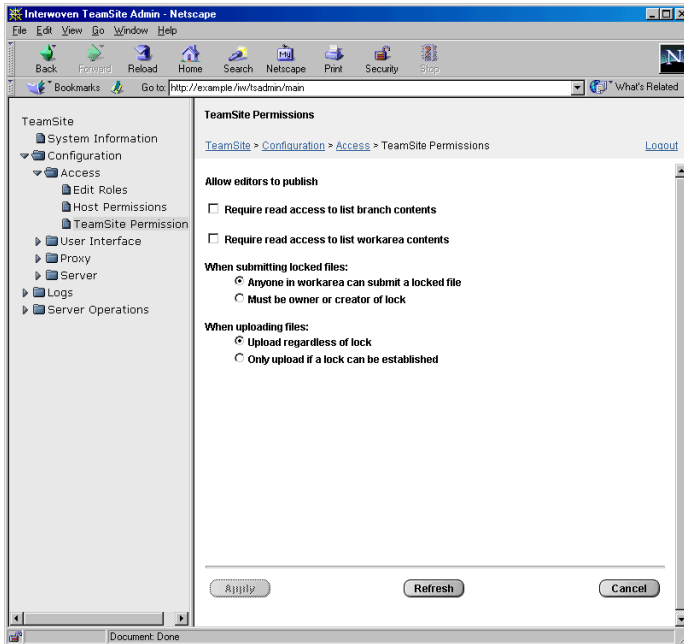
To configure this list manually, see "Domains to Use for Group Authentication" on page 145.

Setting TeamSite Permissions

You can specify TeamSite permissions for:

- Whether Editors can publish.
- Whether users can see the names of branches and workareas where they do not have access.
- Whether anyone, or only the owner or creator of the lock, can submit locked files.
- Whether you want files uploaded only if TeamSite can establish a lock.

To modify TeamSite permissions, navigate to **TeamSite > Configuration > Access > TeamSite permissions**.



TeamSite Permissions window, displaying the default settings

- **Allow editors to publish**—Applies to all Editors on all branches.
To configure this option manually in `iw.cfg`, see page 118.
- **Require read access to list branch contents**—Specifies whether users can see the names of branches where they do not have access.
- **Require read access to list workarea contents**—Specifies whether users can see the names of workareas where they do not have access.

Note: If unchecked, all branch names and workareas appear in the TeamSite GUI, although the branches and workareas where the user does not have read access are not hyperlinked.

To configure these options manually in `iw.cfg`, see page 145.

- **When submitting locked files**—Specifies whether anyone, or only the owner or creator of the lock, can submit locked files.

To configure this option manually in `iw.cfg`, see page 144.

- **When uploading files**—Specifies whether you want files uploaded only if TeamSite can establish a lock.

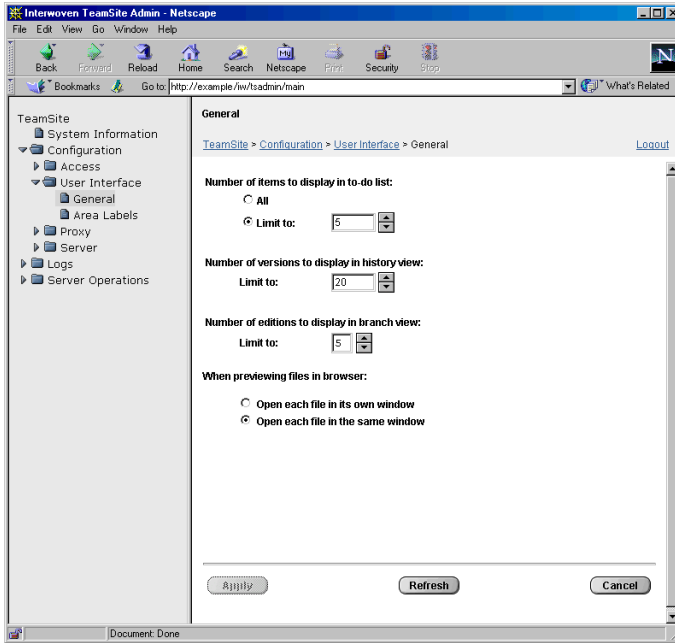
To configure this option manually in `iw.cfg`, see page 135.

Configuring General TeamSite GUI Preferences

You can configure a number of TeamSite GUI elements:

- Number of tasks to display in a To Do or Task list.
- Number of versions to display in the history view.
- Number of editions to display in the branch view.
- How to display files when previewing.

To set general aspects of the TeamSite GUI, navigate to **TeamSite > Configuration > User Interface > General**.



General preferences window, displaying the default settings

- **Number of items to display in to-do list**—Specifies how many jobs to display in an end-user's Task List.

To configure this option manually in `iw.cfg`, see page 135.

- **Number of versions to display in history view**—Limits the number of versions that display in the History view.

To configure this option manually in `iw.cfg`, see page 117.

- **Number of editions to display in branch view**—Limits the number of branches that display in the branch view.

To configure this option manually in `iw.cfg`, see page 116.

- **When previewing files in browser**—Specifies whether previewed files open in their own browser window, or in the same browser window.

To configure this option manually in `iw.cfg`, see page 123.

Changing Area Labels in the TeamSite GUI

You can change the labels that appear in the branch view of the TeamSite GUI, but you should use these options with extreme caution. TeamSite documentation and Interwoven Knowledge Base use the terms “branch,” “workarea,” “staging area,” and “edition” extensively. Changing these labels may confuse users.

To change area labels, navigate to **TeamSite > Configuration > User Interface > Area Labels**.

- **Branch**—Specifies the label for the section of the branch view that lists sub-branches of the main branch.
- **Staging**—Specifies the label for the section of the branch view that displays the staging area.
- **Edition**—Specifies the label for the section of the branch view that lists editions.
- **Work Area**—Specifies the label for the section of the branch view that lists workareas.

To configure these items manually in `iw.cfg`, see page 114.

Configuring the General Proxy Settings

In the General window, you can specify the host name and port number for the following servers:

- **Web Daemon**—Enables secure remote access to TeamSite.

For details about the TeamSite Web daemon, or for instructions on how to configure it manually, see “Configuring the TeamSite Web Daemon and Proxy Server” on page 164.

- **Proxy Server**—Enables virtualization of the Web sites.

For details about the TeamSite proxy server, or for instructions on how to configure it manually, see “Configuring the TeamSite Web Daemon and Proxy Server” on page 164.

- **Content Web server**—Serves the content of the Web sites.

For details about Web servers, or for instructions on how to configure them manually, see “Configuring Web Servers” on page 40.

- **Servlet Engine**—Serves the Java based portions of the TeamSite GUI.

For details about the TeamSite servlet engine, or for instructions on how to configure it manually, see “Servlet Engine” on page 143.

Note: Changes made through the General window affect only the corresponding sections in `iw.cfg`.

To change a setting:

1. Navigate to **TeamSite > Configuration > Proxy > General**.
2. In the **Host** field, enter the name of the server (for example, **example.com**).
3. In the **Port** field, enter the port number of the server.

Note: The Web daemon requires an HTTPS Port. This is the port number of your system’s secure socket layer (SSL), which is usually 443.

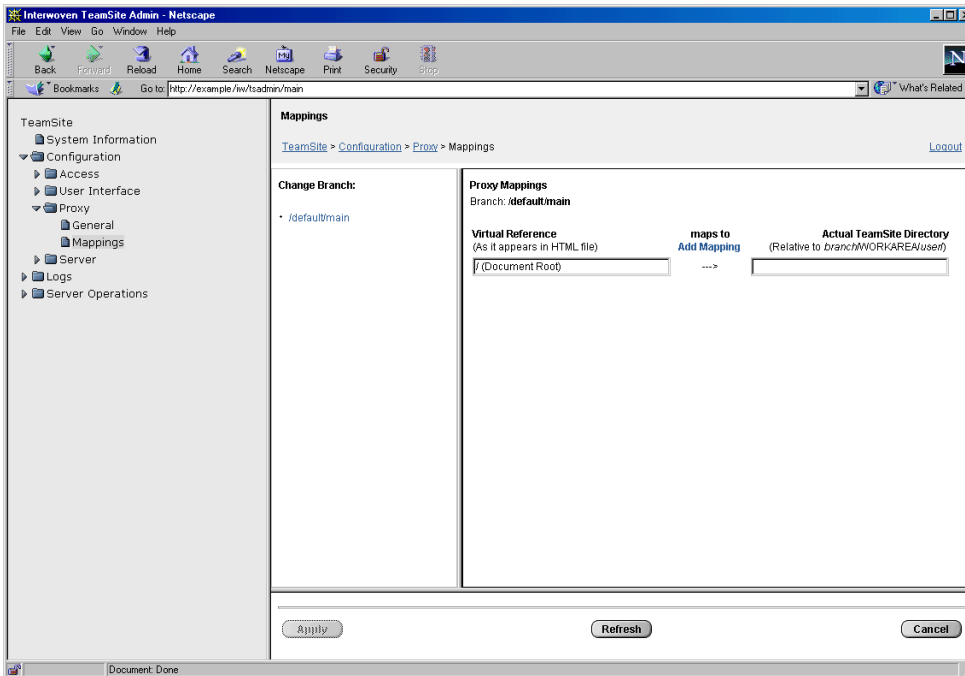
4. Click **Apply**.

Configuring Proxy Mappings

In the Mappings window, you can specify the document root on a branch-by-branch basis, and can create specific directory mappings.

To change the document root of a branch:

1. Navigate to **TeamSite > Configuration > Proxy > Mappings**.



Mappings window

2. In left pane, select a branch.
3. In right pane, in the **Actual TeamSite Directory** field next to / (DocumentRoot), enter the new document root.

To map a directory to a specific directory (other than the new document root):

1. Click **Add Mapping**.
2. Enter the reference as it would appear in the HTML file under **Virtual Reference**.
3. Enter the location it maps to, relative to the top of the user's workarea, under **Actual Teamsite** directory.
4. Click **Apply**.

To remove a specific mapping, click the **Remove** button for that mapping.

For details on the proxy server, or to configure mappings manually, see the section “Configuring the TeamSite Web Daemon and Proxy Server” on page 164.

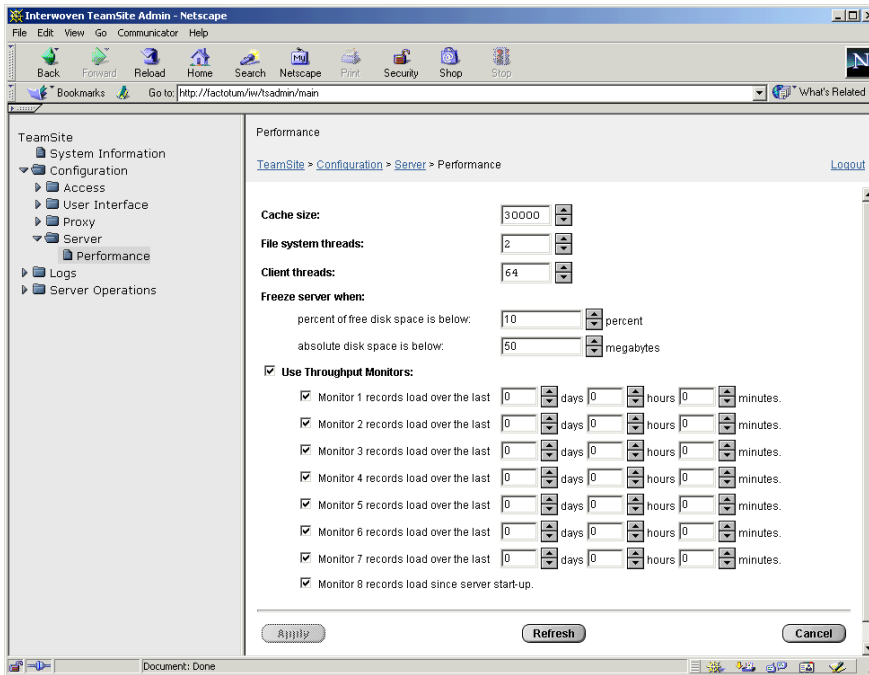
Configuring Server Performance

In the Performance window, you can specify:

- Cache size
- Number of file system threads
- Number of client threads
- Conditions under which you want to freeze the server
- How long you want various throughput monitors to record load

Changes to the cache size, the number of file system and client threads, and throughput monitors take effect only after TeamSite is restarted.

To configure server performance, navigate to **TeamSite > Configuration > Server > Performance**.



The screenshot shows the 'Performance' configuration window in the Interwoven TeamSite Admin interface. The left sidebar contains a tree view with 'TeamSite' expanded, showing 'System Information', 'Configuration', 'Access', 'User Interface', 'Proxy', 'Server', 'Performance', 'Logs', and 'Server Operations'. The 'Performance' section is selected. The main content area displays the following settings:

- Cache size:** 30000
- File system threads:** 2
- Client threads:** 64
- Freeze server when:**
 - percent of free disk space is below: 10 percent
 - absolute disk space is below: 50 megabytes
- Use Throughput Monitors:** (checked)
 - Monitor 1 records load over the last 0 days 0 hours 0 minutes.
 - Monitor 2 records load over the last 0 days 0 hours 0 minutes.
 - Monitor 3 records load over the last 0 days 0 hours 0 minutes.
 - Monitor 4 records load over the last 0 days 0 hours 0 minutes.
 - Monitor 5 records load over the last 0 days 0 hours 0 minutes.
 - Monitor 6 records load over the last 0 days 0 hours 0 minutes.
 - Monitor 7 records load over the last 0 days 0 hours 0 minutes.
 - Monitor 8 records load since server start-up.

At the bottom of the window are buttons for 'Apply', 'Refresh', and 'Cancel'.

Performance window, displaying the default settings

- **Cache size**—Specifies the size of the TeamSite cache. The initial cache size setting should be approximately three times the number of files and directories on the largest branch. For example, if the largest branch contains 15,000 files and directories, you should set cache size to 45000. Maximum cache size is 200,000. If your system's RAM is large enough (2 gigabytes or more), setting cache size to 200,000 can substantially improve performance for operations such as submit.

Caution: Data corruption can occur if the cache size is set to more than the memory can accommodate. For more information about cache size, and configuring cache size manually, see page 155.

- **File system threads**—Specifies the number of file system threads. The value should be set to approximately the number of CPUs on the TeamSite server.

Note: To configure file system threads manually, see page 156.

- **Client threads**—Specifies the number of simultaneous requests TeamSite can handle through the TeamSite GUI or command-line tools. These requests are very short-lived, so that threads are quickly freed for other users. If all threads are in use, TeamSite serializes requests. The default value for this setting is 64, and it should not be altered.

Note: To configure client threads manually, see page 156.

- **Freeze server when**—Specifies the conditions under which you want to freeze the TeamSite backing store.

For details about freezing the backing store, or to configure this feature manually, see page 157.

- **Throughput Monitors**—Specifies the monitors you want to activate and their monitoring intervals. For each of the monitors 1-7, you can specify a time interval in days, hours, or minutes, when you want that monitor to record the load on the server.

To configure throughput monitors manually, see page 157.

Configuring TeamSite Log Files

In the log files Settings window, you can specify:

- The type of information recorded in the Server, Event, and Trace log files.
- The number of events recorded in the Server, Event, and Trace log files.
- How the Windows NT event viewer classifies these TeamSite events: Startup, Shutdown, Freeze, Thaw, Disk Low.

The following table describes the contents of each log file:

Log	Contents
Server	Records the state of TeamSite over time. Tracks when the TeamSite server is started, shut down, mounted, and so on.
Event	Records activities on TeamSite. Tracks when files are submitted, published, branches created, and so on, including DiskLow, Freeze, ShutDown, StartUp, and Thaw events.
Trace	Records irregularities on the TeamSite server. Used by Interwoven Client Services to diagnose system performance and other issues.

To configure log files, navigate to **TeamSite > Logs > Settings**.

- **Number of events logged in update or submit**—Specifies the number of Submit and Get Latest operations to log for workareas. For example, if the value is set to 60, the Event log will contain the 60 most recent Submit or Get Latest operations (as opposed to the 60 most recent files that were submitted or updated).
- **Record individual file details during submit**—Lists, in the submit entries of the Event log, all the files in new or deleted directories.
- **Record individual file details during update**—Lists, in the submit entries of the Event log, all the files in new or deleted directories.
- **Record full list of users and roles in trace log on startup**—Lists all your TeamSite users and their roles in the Trace log during startup.

Note: Be careful when using this feature. If it is left on for too long, your log files will grow extremely large.

- **Windows NT Event Viewer**—Classifies the following TeamSite events as either Success, Failure, Error, Information, or Warning:
 - Startup
 - Shutdown
 - Freeze
 - Thaw
 - Disk Low

Viewing TeamSite Log Files

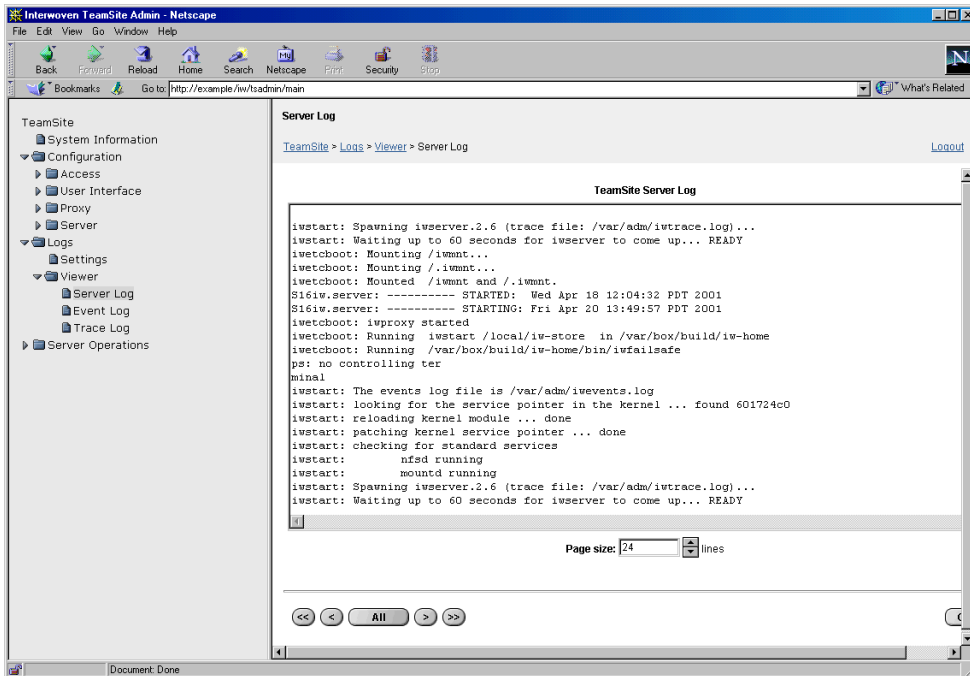
You can view the Server, Event, and Trace log files using the GUI. The following table describes the contents of each of these log files. For convenience, the default location of each log file is also listed.

Log	Location	Contents
Server	<i>iw-home</i> \local\logs\iwserver.log	Records the state of TeamSite over time. Tracks when the TeamSite server is started, stopped, mounted, and so on.
Event	<i>iw-home</i> \local\logs\iwevents.log	Records activities on TeamSite. Tracks when files are submitted, published, branches created, and so on, including DiskLow, Freeze, ShutDown, StartUp and Thaw events.
Trace	<i>iw-home</i> \local\logs\iwtrace.log	Record of any irregularities on the TeamSite server. Used by Interwoven Client Services to diagnose system performance issues.

To view a TeamSite log file:

1. Navigate to **TeamSite > Logs > Viewer** and select the log you want to view.

The following is an example of the TeamSite server log:



LogViewer, displaying the Server log

2. Set the page size by entering the number of lines you want to view at any given time or click **All** if you want the entire contents of the log displayed.

Note: TeamSite servers that have been running for a long time might have extremely large log files. Thus, if you click **All** to view the entries for such a log file, it may take several moments for all the entries to load.

3. Navigate through the log file by clicking <<, <, >, or >>.
 - Click > to move one page toward the most recent entry (the end of the log file).
 - Click < to move one page toward the oldest entry (the beginning of the log file).
 - Click >> to move to the last page of the log file (the most recent entries).
 - Click << to move to the first page of the log file (the oldest entries).

Performing Server Operations

The Server Operations window enables you to perform the following operations:

- Abort
- Freeze and Unfreeze
- Reset

The procedures associated with these operations are described in the following sections.

Abort

To abort a server operation:

1. Navigate to **TeamSite > Server Operations > Abort**.

Currently active operations display in the **Select an operation to abort** field.

2. Select the operation you want to abort.
3. Click **Apply**.

The operation is terminated at the earliest possible time.

4. Click **Refresh** to refresh the list of currently active operations.

Freeze or Unfreeze

To freeze the TeamSite server:

1. Navigate to **TeamSite > Server Operations > Freeze/Unfreeze**.
2. Select **Freeze**.
3. Enter the number of seconds you want the TeamSite server to be frozen.
4. Select if you only want the freeze to operate on batch jobs.
5. Click **Apply**.

To unfreeze the TeamSite server, click the **Unfreeze** option, then click **Apply**.

The freeze and unfreeze operations do not enable you to specify a backing store—they operate on all stores (freezes or unfreezes all of them at a time). If you want to specify a specific store in a MultiStore environment, use the `iwfreeze` CLT. For more information about the `iwfreeze` command-line tool, consult the *TeamSite Command-Line Tool Reference*.

Reset

This operation tells the server to reread its configuration files (equivalent to the `iwreset` command). For more information on the `iwreset` command-line tool, consult the *TeamSite Command-Line Tool Reference*.

To reset the TeamSite server:

1. Navigate to **TeamSite > Server Operations > Reset**.
2. Click **Reset Server**.

Chapter 5

Configuring the TeamSite Server

Most of the settings for the TeamSite server are configured in the main configuration file, `iw-home\etc\iw.cfg` (default location).

Some settings are configured in the following files:

- `iw-home\local\config\submit.cfg`
- `iw-home\local\config\autoprivate.cfg`
- `iw-home\local\config\iwtemplates.cfg`
- `iw-home\local\config\file_encoding.cfg`

Changes to most of these configuration options take effect within a few minutes (although for options that affect the TeamSite GUI, you may have to clear your browser cache in order to see the changes). For these options to take immediate effect, use the `iwreset.exe` command-line tool (CLT). Configuration options that require TeamSite to be restarted in order to take effect are marked throughout this chapter.

For workflow-related configuration options, see the *TeamSite Workflow Developer's Guide*.
For TeamSite Templating configuration, see the *TeamSite Templating Developer's Guide*.

Option	Configuration file	Page
Configuring GUI appearance		
Configuring TeamSite area labels	<code>iw.cfg</code>	page 114
Configuring the number of displayed editions	<code>iw.cfg</code>	page 116
Configuring the number of displayed versions	<code>iw.cfg</code>	page 117
Individual user home page settings	Entity database	page 117
Configuring GUI functionality		
Enabling/disabling Editor publish capability	<code>iw.cfg</code>	page 118

Option	Configuration file	Page
Selectively enabling or disabling SmartContext Editing	iw.cfg	page 118
Adding edit and assign task links to Web pages (Casual Contributor interface)	iw.cfg	page 119
Setting the default LaunchPad interface	iw.cfg	page 121
Setting unique server names for LaunchPad to recognize	iw.cfg	page 121
Configuring domain lists in the login screen	iw.cfg	page 122
Setting the login authentication expiration	iw.cfg	page 122
Setting the number of GUI preview windows	iw.cfg	page 123
Adding custom menu items	iw.cfg	page 125
Configuring submit button behavior	iw.cfg	page 131
Disabling menu items	iw.cfg	page 132
Disabling directory operations	iw.cfg	page 134
Disabling unlocked file auto-upload	iw.cfg	page 135
Setting the number of jobs listed in the To Do List	iw.cfg	page 135
Configuring job attributes and filters	iw.cfg	page 136
Configuring email settings	iw.cfg	page 137
Configuring server functionality		
Setting the encoding of iw.cfg	iw.cfg	page 138
Setting user and role authentication using LDAP	iw.cfg	page 138
Using Domain Local Groups to share workareas	iw.cfg	page 142
Setting the webserver group	iw.cfg	page 142
Configuring the Web daemon	iw.cfg	page 142
Configuring the servlet engine	iw.cfg	page 143
Setting the main branch locking model, owner and group	iw.cfg	page 143
Configuring submit capabilities on locked files	iw.cfg	page 144
Configuring the events logged in the submit and update logs	iw.cfg	page 144
Setting branch and workarea security	iw.cfg	page 145

Option	Configuration file	Page
Configuring which domains to use for group authentication	iw.cfg	page 145
Configuring user and group logging	iw.cfg	page 146
Setting TeamSite file locations	iw.cfg	page 146
Configuring Autoprivate	autoprivate.cfg	page 148
Configuring New File templates	iwtemplates.cfg	page 151
Configuring use of the proxy server	iw.cfg	page 153
Configuring the TeamSite server locale	iw.cfg	page 154
Configuring encoding rules for text files	file_encoding.cfg	page 269
Configuring server performance		
Tuning performance during backing store freezes	iw.cfg	page 155
Setting cache size	iw.cfg	page 155
Setting RPC thread count	iw.cfg	page 156
Setting file system threadcount	iw.cfg	page 156
Setting file system active area cache	iw.cfg	page 156
Configuring throughput monitors	iw.cfg	page 157
Detecting low disk space and inodes	iw.cfg	page 157
Configuring submit filtering		page 158
Changing file attributes at submit time	submit.cfg	page 158
Configuring the TeamSite proxy server		page 164
Configuring proxy server operation	iw.cfg	page 166
Resolving relative and absolute URLs	iw.cfg	page 167
Resolving fully-qualified URLs	iw.cfg	page 171
Redirecting TeamSite views to different areas	iw.cfg	page 175
Configuring TeamSite to use different webserver	iw.cfg	page 178
Configuring external remappings	iw.cfg	page 179

Option	Configuration file	Page
Host header remappings	<code>iw.cfg</code>	page 180
Configuring SSI remappings	<code>iw.cfg</code>	page 181
Configuring proxy failover	<code>iw.cfg</code>	page 181
Configuring TeamSite Embedded Failsafe		page 184
Disabling Embedded Failsafe	<code>iw.cfg</code>	page 184

Configuring GUI Appearance

Configuring TeamSite Area Labels

You can change the labels that appear in the TeamSite GUI (WebDesk Pro) branch view by editing the area label lines in `iw.cfg`. Use this option with caution, however, because the “branch,” “staging area,” “edition,” and “workarea” terms are used throughout TeamSite documentation and in the Interwoven Knowledge Base, and changing these labels may cause confusion among users.

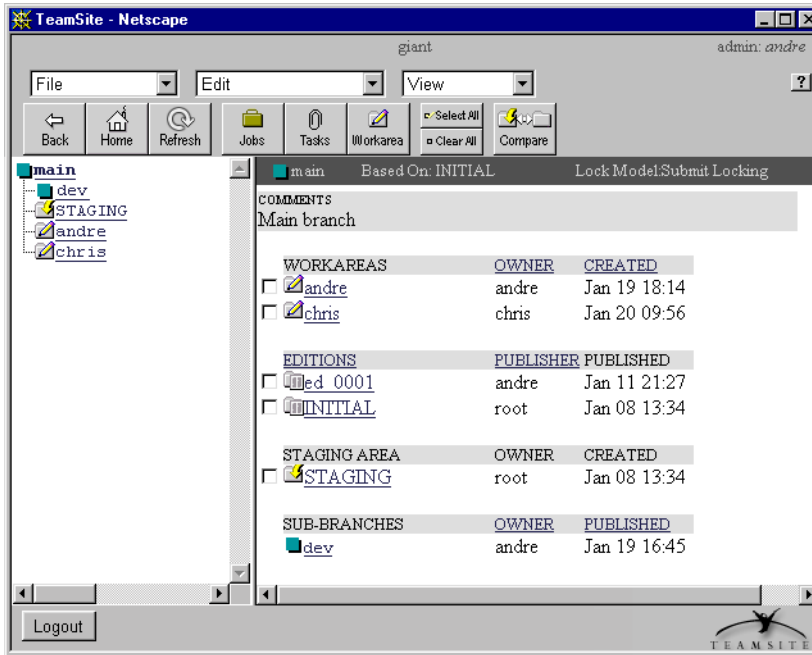
To change these labels, edit the following lines in the `[iwcgi]` section of `iw.cfg`. If these lines do not appear in `iw.cfg`, add them as shown below:

```
branch_label=new_branch_label
staging_label=new_staging_area_label
edition_label=new_edition_label
workarea_label=new_workarea_label
```

For example, with the default values of:

```
branch_label=SUB-BRANCHES
staging_label=STAGING AREA
edition_label=EDITIONS
workarea_label=WORKAREAS
```

The branch view looks like:

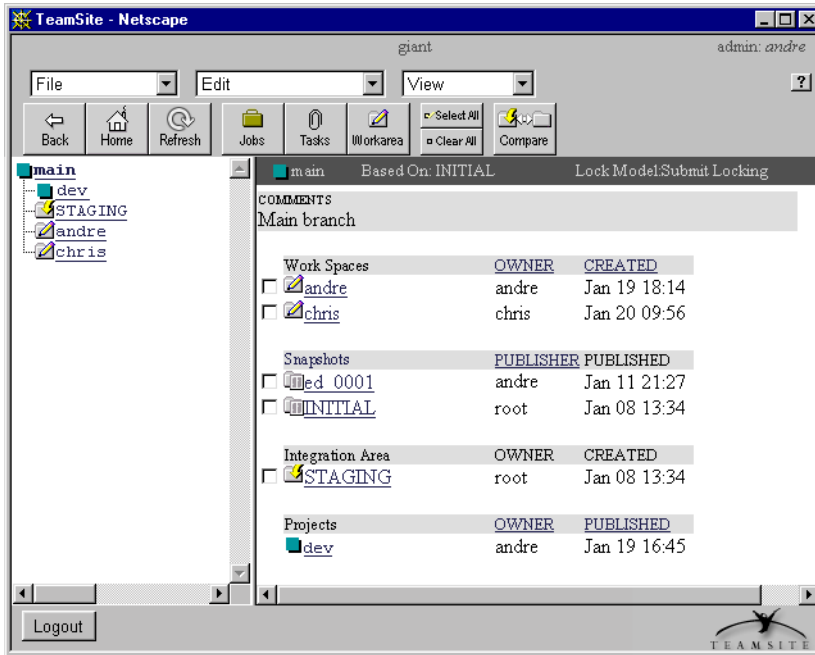


Default TeamSite labels

However, giving these labels other values, such as:

```
branch_label=Projects  
staging_label=Integration Area  
edition_label=Snapshots  
workarea_label=Work Spaces
```

Would change the labels in the branch view to:



The TeamSite Area Labels feature allows you to change the labels that appear in the WorkWindow in the branch view

Configuring Edition Views

You can configure the number of editions you want to see in the branch view of the GUI. To view prior editions, click the **Show all editions** link in the GUI.

To set the number of editions to display, edit the edition list line in the [iwcgi] section of iw.cfg, as shown below:

```
edition_list_limit=number_of_editions
```

For example:

```
edition_list_limit=10
```

would configure TeamSite to display only the ten most recent editions.

If this line does not appear in `iw.cfg`, add it as shown above. The default value is 5 (by default, TeamSite will display the five most recent editions in the branch view).

To show all editions by default, comment out the `edition_list_limit` line by adding a `#` to the beginning of the line.

Configuring History Views

You can configure the number of versions shown in the History view of the TeamSite GUI. To configure this option, use the `view_history_limit` parameter in the `[iwcgi]` section of `iw.cfg`. For example:

```
view_history_limit=5
```

would restrict the History view to show only the five most recent versions of a file. All versions would still exist; however, only five would be displayed.

User Profiles

The `SetHomePage` functionality (where users can set their Home page) in the WebDesk Pro GUI now stores the homepage information in the entity database instead of the `iwprofiles` directory.

If you are upgrading to TeamSite 5.5.1, you must run the `iwprefconv` CLT (as described in the *Command-Line Tools* manual) once to copy any existing homepage information from the `iwprofiles` directory (`iw-home/local/iwprofiles`) to the entity database (`iw-home/local/entities/data`).

Do not modify these files manually. These files are automatically generated by the TeamSite server and updated as needed.

Configuring GUI Functionality

Disabling Editor Publish Capability

TeamSite allows you to turn off Editors' ability to publish. You cannot turn this option off for selected Editors; it applies to all Editors on all branches.

To turn off the Publish capability for Editors:

If applicable, remove the comment mark (#) from the `editor_publish` line in the `[main]` section of `iw.cfg`. If `iw.cfg` does not contain this line, add it as shown below.

```
editor_publish=no
```

Enabling and Disabling SmartContext Editing

You can selectively enable or disable SmartContext Editing for different workareas or files by adding lines to the `[iwproxy_smartcontextedit_allowed]` section of `iw.cfg`. If this section does not exist, SmartContext Editing is enabled by default.

The `[iwproxy_smartcontextedit_allowed]` section contains one `_default` line, which specifies whether SmartContext Editing is turned on or off in any area or for any file not otherwise specified. This section can also contain any number of `_regex` lines. Each `_regex` line uses a case-insensitive regular expression to specify areas or files, and then specifies whether SmartContext Editing is enabled or disabled for the specified items. A `_regex` line has the following case-insensitive syntax:

```
_regex=regular-expression=yes|no
```

`_regex` lines are order-dependent. For example, the following `[iwproxy_smartcontextedit_allowed]` section turns SmartContext Editing on by default, and it explicitly turns it on for all files in all of Andre's workareas on all branches. It then turns SmartContext Editing off for all CGI files. Because the line turning SmartContext Editing on for

Andre's workareas comes first, he will be able to use SmartContext Editing for CGI files in his workarea:

```
[iwproxy_smartcontextedit_allowed]
_default=yes
_regex=(.*)/WORKAREA/andre/.*=yes
_regex=\.cgi(\?.*)?$=no
```

The Casual Contributor Interface: Adding Editing and Task Links to Web Pages

You can give Authors the ability to access WebDesk file editing and task features directly from any Web page or email message by adding one or more URL links to the source file or message. When a user clicks on one of these links, the user will be taken directly to the appropriate functionality within TeamSite.

If the user is not already authenticated, the TeamSite login screen is displayed.

To create a link, use the following syntax:

`http://servername/iw/webdesk/function?vpath=filename`

or

`http://servername/iw/webdesk/taskaction?taskid=taskid`

where the variables are defined as follows:

<i>servername</i>	The name of the server
<i>function</i>	One of: <ul style="list-style-type: none"> • <code>assign</code>—prompts the user to create a new task using <code>default_assign.wft</code> with the specified file attached. • <code>edit</code>—opens the file for editing in WebDesk. • <code>sce</code>—opens the specified file in a browser for use with SmartContext Editing. • <code>tag</code>—opens the Metadata Capture dialog. If MetaTagger 3.0 is installed, this will launch its Metadata Capture dialog. • <code>details</code>—displays the File Properties window. • <code>visualdiff</code>—displays the Visual Difference window, comparing the specified file with the version in the staging area.
<i>filename</i>	Directory path and filename of the file.
<i>taskaction</i>	One of: <ul style="list-style-type: none"> • <code>task</code>—displays the corresponding task ID. • <code>transitiontask</code>—opens the Task Transition dialog. • <code>taketask</code>—grants the user ownership of the specified group task (<i>taskid</i> must refer to a group task).
<i>taskid</i>	Integer ID of a task.

For example, if you want to create a link from a Web page that opens the source file `sample.html` in Edit mode, from the server `example`, and the source file is in `/default/main/example/sample.html`, you would enter the following URL in the link:

```
http://webdev/iw/example/edit?vpath=/default/main/dev/sample.html
```

Sample HTML including the above link might look something like this:

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
<TITLE>TeamSite URL Example</TITLE>
```



```

</HEAD>

<BODY>
<H1>TeamSite URL Example</H1>
<P>
With the TeamSite WebDesk user interface, it is easy for users to
<A HREF="http://example/iw/webdesk/edit?vpath=/default/main/dev/
sample.html">edit</A> any file in your asset base!
</P>
</BODY>
</HTML>

```

You can also embed a task link within an email message or Web page by using the following syntax:

```
http://servername/iw/webdesk/task?taskid=number
```

where *servername* is the name of the server and *number* is the task ID.

Setting the Default LaunchPad Interface

TeamSite offers both an applet and a standalone application version of LaunchPad. The entry in the [iwcgi] section of iw.cfg is set to use the applet by default:

```

[iwcgi]
use_launchpad_applet=true

```

If you prefer for all clients to use the LaunchPad application, change the use_launchpad_applet setting to false.

Note that clients who use Netscape on the Macintosh or who have Java turned off in their browsers cannot use the applet. They will always use the application.

Setting Unique Server Names for LaunchPad to Recognize

To enable LaunchPad to differentiate between different TeamSite servers, set a unique server name in iw.cfg:

```
launchpad_hostname=hostname
```

where *hostname* is the server name.

Configuring Domain Lists in the Login Screen

To configure which domain lists appear in the Domain pull-down menu of the TeamSite login screen, configure the `domain_list` option in the `[iwcgi]` section of `iw.cfg`. To configure this option:

1. If a comment symbol (#) is present at the beginning of the `domain_list` line in the `[iwcgi]` section of `iw.cfg`, remove it. Do not confuse this line with the `domain_list` line in the `[iwserver]` section of `iw.cfg`. If `iw.cfg` does not contain this line, add it as shown below.
2. Edit the line to read:

```
domain_list=domain1,domain2,domain3
```

You can include any number of domains in this list.

Setting Login Authentication Expiration

If a user logs in to WebDesk, then closes the window, then opens WebDesk again within the same user session on a machine within a 24-hour period, TeamSite recognizes the user from his previous login and does not require him to re-enter a username and password. This is important for users of the Casual Contributor interface, who will be transferred directly to the relevant part of WebDesk if they have previously logged in within the authentication period.

The default login authentication expiration is 24 hours. To change this value, edit the `ui_login_lifetime` setting in the `[authentication]` section of `iw.cfg` using the syntax:

```
ui_login_lifetime=ahbmcs
```

or

```
ui_login_lifetime=xYyMzD
```

where *a* represents the number of hours, *b* represents the number of minutes, and *c* represents the number of seconds; or *x* represents the number of years, *y* represents the number of months,

and z represents the number of days. Do not enter spaces in the value. Note that if you are specifying years, months, and days that Y, M, and D must be upper-case.

For example, to set the login authentication expiration to 8 hours, enter the value 8h, as shown below:

```
[authentication]
ui_login_lifetime=8h
```

To set the login authentication expiration to 8 hours, 15 minutes, and 7 seconds, enter the values 8h15m7s, as shown below:

```
[authentication]
ui_login_lifetime=8h15m7s
```

To set the login authentication expiration to 1.5 years, enter the values 1y6m, as shown below:

```
[authentication]
ui_login_lifetime=1y6m
```

If you want the login authentication to never expire, enter `infinite`, as shown below:

```
ui_login_lifetime=infinite
```

Configuring Preview Windows

When you click on the name of a file in the TeamSite GUI, TeamSite launches a browser window so you can preview it. By default, a new browser window is launched each time you click on a file name. However, you can configure TeamSite to reuse the same window each time you preview a file.

To configure the number of browser windows TeamSite launches, you must edit the `single_browser_window` line in the `[iwcgi]` section of `iw.cfg`. If this line does not exist, add it as shown below.

To reuse the same window each time you preview a file, set `single_browser_window=TRUE`. To launch a new browser window each time you preview a file, set `single_browser_window=FALSE` or comment the line out altogether. This setting will apply to all users on all branches of the TeamSite server.

```
[iwcgi]  
single_browser_window=TRUE
```

Custom Menu Items

You can add custom menu items to either WebDesk or WebDesk Pro. These menu items can call either CGI scripts or HTML pages.

Note that TeamSite includes all custom **File** menu items in the **File Options** drop-down list for each file listed in the Task Details screen of WebDesk Pro. TeamSite checks `iw.cfg` for **File** custom menu items, and adds them to the **File Options** drop-down list. Custom menu items for other menus (**Edit** and **View**) are not included in the Task Details screen in WebDesk Pro.

About CGI Scripts

CGI scripts that are added to the TeamSite interface are executed via the TeamSite CGI launcher/wrapper, which calls the CGI program and sets the environment variables that would be set by the webserver.

Enabling Custom Scripts in the TeamSite GUI

To allow CGI scripts written in Perl to be executed directly from the TeamSite GUI, you must execute a wrapper script that calls on the Perl script to be executed. For example, to extend TeamSite drop-down menu items to allow users to execute the Perl script `custom_script.cgi`, you would do the following:

1. Create a wrapper script, `custom_script.bat`, and place it in the `iw-home\httpd\iw-bin` directory. Here is a sample wrapper script:

```
@echo off
iw-home\iw-perl\bin\iwperl iw-home\httpd\
iw-bin\custom_script.cgi
```

2. Edit `iw.cfg` to add the custom menu item by referencing `custom_script.bat`:

```
[iwcgi]
custom_menu_item_unique_identifier="Menu", "Name",
"custom_script.bat", "RolesList", "WindowAttributes"
```

Your TeamSite server will now be configured to properly execute your custom written Perl scripts.

Creating Custom CGI Scripts

The CGI wrapper makes certain variable=value pairs available, depending on the user's current location and any items that are selected.

The user's username and role are always available. Also available is the vpath of the current location, the mount path of the TeamSite mount point, and the directory paths, object ids, and names of the current branch and archive, and (if applicable) of the current sub-branch, area, and directory. The `page_type` variable indicates what type of TeamSite area the user is currently in, and the `subpage_type` variable (available when the user is not on a branch page), indicates whether the user is currently in a sub-directory or the root directory of his current area. Each item selected has four variables associated with it: type, object id, name, and path.

For example, user `andre` logged in as Master might navigate into the `htdocs` directory in his workarea on the `main` branch, and select the checkboxes next to two directories. The following variable=value pairs would then be available (see next page):

iw_prog_name=custom_script.cgi	—	Name of the CGI script
wrapper_version=1	—	Version of the wrapper
domain=CHOCOLATE	—	Windows domain
vpath=/default/main/WORKAREA/andre/htdocs	—	Vpath of current location
mount_path=/iwmnt	—	TeamSite mount point
directory_id=0x0000007b00000079000000b9	}	Attributes of current directory
directory_name=htdocs		
directory_path=/iwmnt/default/main/WORKAREA/andre/htdocs		
area_id=0x000021000000000000000007b	}	Attributes of current area
area_name=andre		
area_path=/iwmnt/default/main/WORKAREA/andre		
branch_id=0x000022500000000000000006d	}	Attributes of current branch
branch_name=main		
branch_path=/iwmnt/default/main		
archive_id=0x0000202000000000000000001	}	Attributes of current archive
archive_name=default		
archive_path=/iwmnt/default		
subpage_type=sub_directory	—	Type of directory
page_type=workarea	—	Type of area
session=AAAAAQAFYw5kcmUAAAKMjAwMS5hbmRyZTWZhj4A	—	For impersonation use
page_id=8	—	For internal use
user_name=andre	}	User's name and role
user_role=master		
type_0=directory	}	Type, object id, name, and directory path for the first item selected
objid_0=0x0000007b000000b9000000e9		
name_0=corporate		
path_0=/iwmnt/default/main/WORKAREA/andre/htdocs/corporate		
type_1=directory	}	Type, object id, name, and directory path for the second item selected
objid_1=0x0000007b000000b9000000e5		
name_1=news		
path_1=/iwmnt/default/main/WORKAREA/andre/htdocs/news		

Adding Custom CGI Scripts to WebDesk and WebDesk Pro

To add a custom CGI script to the TeamSite GUI (WebDesk or WebDesk Pro):

1. Create a CGI program in *iw-home\httpd\iw-bin*.
2. Add the following line to the `[iwcgi]` section of `iw.cfg`:

```
custom_menu_item_identifier="MenuName", "MenuItemName",
"CGIProgramName", "RolesList", "WindowAttributes",
"WindowName", "500"
```

where *identifier* is a unique identifier for the menu item, and the parameters are as follows. Note that some parameters differ depending on whether you want the menu item to appear in WebDesk or WebDesk Pro:

Parameter	Description	WebDesk	WebDesk Pro
<i>MenuName</i>	The menu to add the entry to.	Required Possible values are Edit or View.	Required Possible values are File, Edit, or View.
<i>MenuItemName</i>	The name of the menu item as it appears to the user.	Required	Required
<i>CGIProgramName</i>	The CGI program to execute (must be in <i>iw-home\httpd\iw-bin</i>). The program name may not contain spaces.	Required	Required
<i>RolesList</i>	The comma-separated list of roles who will have access to this menu item.	Required Must contain author or all.	Optional ¹ (required if <i>WindowAttributes</i> is specified) Can contain author, editor, admin, master, or all.

Parameter	Description	WebDesk	WebDesk Pro
<i>WindowAttributes</i>	Specifies the characteristics of the window.	Required See table below for possible attributes.	Optional ² (required if <i>WindowName</i> is specified) See table below for possible attributes.
<i>WindowName</i>	Specifies the name of the window. If the menu item does not need a window, specify <i>_nowindow</i> .	Required	Optional
500	Specifies that this menu item will appear in WebDesk. Menu items that appear in WebDesk will also appear in WebDesk Pro.	Required	Omit if you want the menu item to appear in WebDesk Pro but not in WebDesk. If included, follow WebDesk requirements.

1. If *RolesList* is not specified, all roles are assumed.
2. If *WindowAttributes* is not specified, the defaults are:
resizable=yes,scrollbars=no,menubar=yes,width=640,height=480

WindowAttributes are specified as follows. Unless otherwise specified, all window attributes are of the form *value*=yes | no.

toolbar	Specifies whether or not the browser toolbar will appear.
location	Specifies whether the Location input field (for entering URLs) will appear.
directories	Specifies whether directory buttons will appear.
status	Specifies whether the status line will appear.
scrollbars	Enables scrollbars.
resizable	Allows the user to resize the window.
menubar	Specifies whether the browser menu bar will appear.
width	Specifies the width of the window (in pixels).
height	Specifies the height of the window (in pixels).

For example:

```
custom_menu_item_show_env="File", "Environment", "show_env.cgi",  
"admin, master", "width=640,height=450,scrollbars=yes,resizable=yes"
```

creates a new custom menu item called `show_env`. This menu item will appear in the **File** menu of WebDesk Pro, and it will be called **Environment**. It will call the CGI program `show_env.cgi`, and the menu item will be available only to Administrators and Master users. The window that appears will be 640 pixels wide by 450 pixels high, it will have scrollbars, and it will be resizable.

```
custom_menu_item_reports="View", "Reports", "report.cgi", "admin,  
master, author", "scrollbars=yes,resizable=yes,width=640,height=545",  
"reports", "500"
```

creates a new custom menu item called `reports`. This menu item will appear in the **View** menu of both WebDesk and WebDesk Pro, and it will be called **Reports**. It will call the CGI program `report.cgi`, and the menu item will be available only to Authors, Administrators and Master users. The window that appears will be 640 pixels wide by 545 pixels high, it will have scrollbars, and it will be resizable. The window will be named `reports`.

3. Log in and select the menu where you added the new item. You will see the new menu item at the bottom of the menu. When you select this item, a separate window will display the output of your CGI program.

Adding HTML Pages to WebDesk and WebDesk Pro

You can also launch custom HTML files in a separate window from TeamSite. The HTML file will be called directly from the web browser, without any special preprocessing.

To add an HTML file to TeamSite (WebDesk or WebDesk Pro):

1. Create the HTML file.
2. Add the following line to the `[iwcgi]` section of `iw.cfg`:

```
custom_menu_item_identifier="MenuName", "MenuItemName", "file:URL",  
"RolesList", "WindowAttributes", "WindowName", "500"
```

where all parameters except `file:URL` are specified as described in “Adding Custom CGI Scripts to WebDesk and WebDesk Pro” on page 128. *URL* is the URL of the HTML file to call. For example, an entry that calls the file `www.example.com/internal/localhelp.html` might look like:

```
custom_menu_item_help="View", "Local help", "file:www.example.com/
internal/localhelp.html"
```

WindowName and 500 are required for adding custom HTML pages to WebDesk. If you only want a page to appear in WebDesk Pro, omit the 500 parameter. In this case, *WindowName* is optional.

Configuring Submit Button Behavior

The **Submit** button can be configured to either submit files directly to the staging area (Submit-Direct), or to use the default Submit workflow process (Submit-Process). By default, the Submit button will use workflow for all roles.

To configure the Submit button behavior on a per-role basis, add the following section to `iw.cfg`:

```
[submit_button]
submit_direct=roleslist
submit=roleslist
```

where *roleslist* specifies the roles that use this option (editor, admin, master, or all—Authors must always use the Submit workflow process).

For example:

```
[submit_button]
submit_direct=admin, master
submit=editor
```

would configure the **Submit** button to submit files directly to the staging area for all Administrators and Master users, but to use the Submit workflow process for all Editors. Authors would still use the Submit workflow process.

If you disable the Submit button for any role (that is, if you include `submit=roleslist` in both the `[ui_remove_menu_items]` section and the `[ui_disable_directories]` section—see

below), then this section will have no effect for that role, as there will be no Submit button to configure.

Disabling Menu Items

You can now disable TeamSite menu items and buttons on a per-role basis. To disable a TeamSite menu item, add a new section to TeamSite's main configuration file, `iw.cfg`, as follows:

```
[ui_remove_menu_items]
menuitemname="roleslist"
```

where *menuitemname* is the name of the menu item you want to disable, and *roleslist* is a comma-separated list of roles (e.g. `author, editor`). The menu item will be disabled in WebDesk Pro for all roles specified, and in WebDesk if `author` is specified. If the menu item has a corresponding button, it will be removed from the WebDesk Pro Button Bar for these roles (the **Compare Any** menu item and the **Compare with Staging** button are both governed by the `compare` value of *menuitemname*).

You can add multiple lines to a `[ui_remove_menu_items]` section. For example, the following `[ui_remove_menu_items]` section turns off the **Delete** and **Move** menu items for Editors and Authors.

```
[ui_remove_menu_items]
delete="editor,author"
move="editor,author"
```

This is a complete list of values of *menuitemname* for the `[ui_remove_menu_items]` section. Items marked with an asterisk (*) only apply if TeamSite Templating is installed:

Value	Disabled Menu Item	Disabled Button
<code>assign</code>	File > Assign	Assign
<code>compare</code>	File > Compare Any	Compare (compares with the staging area)
<code>copy</code>	File > Copy	N/A
<code>copyto</code>	File > Copy to Area	N/A
<code>delete</code>	File > Delete	N/A

Value	Disabled Menu Item	Disabled Button
*editdcr	Edit > Edit Data Record	N/A
editfile	Edit > Edit File	Edit File
editfilewith	Edit > Edit File With	N/A
file_properties	File > File Properties	N/A
*genHTML	File > Generate HTML	N/A
getlatest	File > Get Latest	Get Latest
history	View > History	N/A
import_files	File > Import Files	N/A
listlocks	View > List Locks	N/A
listmodified	View > List Modified	N/A
lock	Edit > Lock	N/A
move	File > Move	N/A
new_branch	File > New Branch	N/A
*newdcr	File > New Data Record	N/A
newdir	File > New Directory	N/A
newfile	File > New File	N/A
newJob	File > New Job	N/A
new_workarea	File > New Workarea	N/A
private	Edit > Private	N/A
public	Edit > Public	N/A
publish	File > Publish	N/A
*regenHTML	File > Regenerate HTML	N/A
rename	File > Rename	N/A
setHomePage	Edit > Set Home Page	N/A
submit	File > Submit	Submit (configurable—see page 131)
submit_direct	File > Submit-Direct	Submit (configurable—see page 131)

Value	Disabled Menu Item	Disabled Button
submitlog	View > Submit Log	N/A
task_todo	View > To Do List	To Do
unlock	Edit > Unlock	N/A
updatelog	View > Update Log	N/A
viewfile	Edit > View File	View File

Disabling Directory Operations

You can now disable certain operations' ability to act on directories in WebDesk Pro, on a per-role basis. To disable a TeamSite menu item's abilities to act on directories, add a new section to the TeamSite main configuration file, `iw.cfg`, as follows:

```
[ui_disable_directories]
menuitemname="roleslist"
```

where *menuitemname* is the name of the menu item you want to disable for directories, and *roleslist* is a comma-separated list of roles (for example, `author, editor`). Specify all to disable the menu item for all roles. The menu item will no longer act on directories when it is invoked by a user who is logged in to WebDesk Pro with one of these roles.

You can only disable menu items' abilities to act on directories if they would ordinarily be able to do so.

You can add multiple lines to a `[ui_disable_directories]` section. For example, the following `[ui_disable_directories]` section disables Editors' and Authors' abilities to delete or move directories.

```
[ui_disable_directories]
delete="editor,author"
move="editor,author"
```

This is a complete list of values of *menuitemname* for the [ui_disable_directories] section:

Value	Disabled Menu Item	Disabled Button
assign	File > Assign	Assign
copy	File > Copy	N/A
copyto	File > Copy to Area	N/A
delete	File > Delete	N/A
getlatest	File > Get Latest	Get Latest
move	File > Move	N/A
private	Edit > Private	N/A
public	Edit > Public	N/A
rename	File > Rename	N/A
submit	File > Submit	Submit (configurable—see page 131)
submit_direct	File > Submit-Direct	Submit (configurable—see page 131)

Disabling Unlocked File Auto-Upload

By default, TeamSite allows you to upload files even if it cannot establish a lock on them. To disable this feature so that files are uploaded only if TeamSite can establish a lock, add the following line to the [iwproxy] section of iw.cfg:

```
allow_unlocked_file_upload=no
```

To turn unlocked file uploading back on, either remove this line from iw.cfg or set it to yes.

Setting the Number of Jobs Listed in the To Do List

The iw.cfg configuration file allows you to set the maximum number of jobs to be listed in the To Do List. To configure this option, add the following line to the [iw_workflow_ui] section of iw.cfg.

If this section does not exist, create it as follows:

```
[iw_workflow_ui]
max_job_count_per_page=number_of_jobs|all
```

The default is 55 jobs per page. You can change the number of jobs by specifying a value. If you specify *all*, all the jobs fulfilling a particular view display on a single page. In Windows 95/98, if you have over 100 jobs and specify *all*, the workflow screen may be exceedingly slow.

Configuring Job Attribute Filters and Settings

Job attributes are configurable properties of individual jobs. These attributes show up in the workflow section of the TeamSite GUI. You can set these attributes through the GUI, and you can choose to display only the jobs that have certain attribute settings. The names of these attributes and their possible settings are configured in the `[iw_workflow_ui]` section of `iw.cfg`.

For example, if you have attributes called Category, Month, and Date, you can use drop-down menus in the GUI to view only jobs with specific values of Category, Month, and Date (for example, Category=Marketing, Month=November, Date=24).

You can configure up to three attributes. Each attribute can have any number of possible values. Values are separated by colons.

The `[iw_workflow_ui]` section of `iw.cfg` has the following format:

```
[iw_workflow_ui]
attribute1=attributename1
values1=valuelist1
attribute2=attributename2
values2=valuelist2
attribute3=attributename3
values3=valuelist2
```

where *attributename1*, *attributename2*, and *attributename3* specify the names of attributes 1, 2, and 3, respectively. *valuelist1*, *valuelist2*, and *valuelist3* are of the format:

```
value1:value2:value3:...valuen
```


To use fewer than three attributes, omit the appropriate attribute and value lines. Attributes must always be numbered sequentially, starting at 1. For example:

```
[iw_workflow_ui]
attribute1=Category
values1=Sales:Marketing:Engineering:Professional Services:Administration
attribute2=Month Due
values2=Jan:Feb:Mar:Apr:May:Jun:Jul:Aug:Sep:Oct:Nov:Dec
attribute3=Date Due
values3=1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24
:25:26:27:28:29:30:31
```

would create three attributes named Category, Month Due, and Date Due. Category has possible values of Sales, Marketing, Engineering, Professional Services, and Administration. Month Due has possible values of Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, and Dec. Date Due can be any number from 1 to 31.

Note: Due to page constraints, some lines may appear to wrap. Actual lines in `iw.cfg` should never wrap.

Configuring Email Settings

The TeamSite Assign feature sends email to the recipient of a task. The following settings in the `[iwsend_mail]` section of `iw.cfg` enable you to configure how this email is sent:

- `maildomain=domain.topleveldomain`
Specifies the domain (for example, `maildomain=example.com`)
- `mailserver=servername.domain.topleveldomain`
Specifies the mail server to use.
- `use_mapping_file=false|true`
Optional entry that specifies whether or not to use a mapping file to configure individual email addresses or aliases.
- `email_mapping_file=path_to_file`
Optional entry that specifies the location of the mapping file to use (a sample file is located in `iw-home\local\config\wft\email_map.cfg`).

Configuring Server Functionality

Specifying the Encoding of the iw.cfg File

To facilitate the internationalization of the TeamSite, you now have the ability to use text editors that save the `iw.cfg` file in various encodings. The encoding setting in the first section of the `iw.cfg` file declares the encoding of the `iw.cfg` file itself. The default setting is `ascii` and specified using the following syntax:

```
[iwcfg]
encoding=ascii
```

Note: This *must* be the first section in the `iw.cfg` file—no other entry can precede it.

You can edit the encoding setting using any of the following values:

Preferred Encoding	Also Valid
<code>ascii</code> (default setting)	<code>us-ascii</code>
<code>ISO-8859-1</code>	<code>iso_8859-1</code> and <code>latin1</code>
<code>windows1252</code>	<code>cp1252</code>
<code>euc-jp</code>	<code>euc_jp</code>
<code>shift_jis</code>	<code>shift-jis</code> , <code>sjis</code> , and <code>cp932</code>
<code>utf-8</code>	<code>utf_8</code> and <code>utf8</code>

Note: You cannot have a space between the equal sign (=) and the encoding type.

User and Role Authentication Using LDAP

User Authentication Using LDAP

TeamSite can be configured to authenticate users using LDAP. To do this, complete the following procedure:

1. Add the following lines to the `[authentication]` section of `iw.cfg`:

```
authenticate_by=ldap
ldap_server=ldap-server
ldap_port=ldap-port
ldap_dnbase=search-base-location
ldap_key=key
```

where:

- *ldap_server* is the name or IP address of the LDAP server
- *ldap_port* is the port for the LDAP server (optional; the default value is 389)
- *search-base-location* is the specification of DN base location according to LDAP search syntax (for example, `ldap_dnbase=ou=people,o=example.com`)
- *key* is the name of the LDAP attribute that holds the user account names (optional)

2. Save and close the file.

For information about creating new attributes in the Active Directory service, refer to:

<http://www.microsoft.com/WINDOWS2000/library/planning/activedirectory/manadsteps.asp>.

Note: If you are using LDAP, you must reset the TeamSite server with the `iwreset` command after you make changes to the LDAP database. This command causes the TeamSite server to replace existing user information in its local cache with new data from the LDAP server.

Role Authentication Using LDAP

TeamSite role information is often stored separately from user authentication data. However, TeamSite enables you to store role information in an LDAP database along with user authentication data. TeamSite supports LDAP user and role authentication using the Windows NT 4.0 domain model, or by using Active Directory on Windows 2000 servers.

Note: Placing TeamSite roles in your LDAP server means that these roles can not be queried or manipulated programmatically via Interwoven's OpenAPI interface. TeamSite roles within your LDAP server can be queried and manipulated directly via your LDAP tool or programmatically via an LDAP server programming interface supplied by other vendors, not by Interwoven.

Every LDAP directory has a schema which describes the objects and attributes that are found in the directory. For example, you could have an object called `user` and an attribute

postaladdress. To configure TeamSite to perform user and role authentication, you can either modify an existing attribute to represent TeamSite roles or create a new one.

For more information about modifying LDAP schemas, see “Modifying LDAP Schemas to Store TeamSite Roles” on page 141.

If you want to use your LDAP database to authenticate TeamSite roles (Author, Editor, Administrator, Master), add the following line to the [authentication] section:

```
ldap_roles=role-attribute-name
```

where *role-attribute-name* is the attribute name from the LDAP schema that stores TeamSite roles.

Additionally, you must modify your LDAP schema as described in “Modifying LDAP Schemas to Store TeamSite Roles” on page 141.

Configuring TeamSite and Active Directory to Work Without Using an Anonymous Bind

In some installations, anonymous binds to Active Directory are not permitted for security reasons. If you cannot use an anonymous bind to Active Directory to read user names, you can establish a dedicated Active Directory user account to use for user authentication. All searches for users’ Distinguished Names will be done using this account instead of an anonymous bind. For this mode of operation, you must add two additional parameters to the [authentication] section of iw.cfg.

```
ldap_account=DistinguishedName  
ldap_pwd=password
```

where:

DistinguishedName specifies the Distinguished Name of the Active Directory user account to be used for Active Directory searches. Note that this is not a simple account name, but a complete Distinguished Name (DN) for a user. For example:

```
ldap_account=cn=TeamSite,cn=Users,dc=myCompany,dc=com.
```

password specifies the clear text password of the Active Directory user account that matches *ldap_account*.

Note that the user name and password are in clear text, so it is important to limit who has read permission for *iw.cfg*. You can change the account name and password at any time. The changes will take effect the next time the server is restarted or reset.

Modifying LDAP Schemas to Store TeamSite Roles

If you do not have an existing attribute in your LDAP schema where you can store TeamSite roles, add a new attribute to your LDAP schema. If you do have an attribute where you can store TeamSite roles, start with step 3.

1. Add an auxiliary class to an existing object in the schema.
2. Add a new attribute to that object named *tsrolesattribute*.
3. Edit the [authentication] section of *iw.cfg* to include:
`ldap_roles=tsrolesattribute`
4. Your LDAP administrator can now assign TeamSite roles (Master, Administrator, Editor, Author) to users configured in your LDAP server using the server's administration tools.

The following are the valid values (they are case sensitive):

- master
- admin
- editor
- author

5. Save and close the file.

Notes:

- Placing TeamSite roles in your LDAP server means that these roles can not be queried or manipulated programmatically via the OpenAPI interface. TeamSite roles within your LDAP server can be queried and manipulated directly via your LDAP tool or programmatically via an LDAP server programming interface supplied by other vendors, not by Interwoven.

- You cannot store TeamSite role information in your LDAP database if you want to use operating system authentication. If you want to store role information in your LDAP database, also use LDAP for authentication.
- For information on modifying schemas and adding attributes for the Netscape Directory Server, refer to the *Directory Server Administrator's Guide* (<http://home.netscape.com/eng/server/directory>).

Using Domain Local Groups to Share Workareas

If you are using native mode Active Directory for authentication for a Windows 2000 TeamSite server, and you want to use Domain Local Groups as the group for sharing a workarea, add the following line to the [iwserver] section of iw.cfg:

```
domain_local_groups=yes
```

By default this option is disabled.

Webserver Group

The webserver group setting should be set to any group that allows the webserver to see the web content as an outside viewer would see it, in order for users to be able to preview the Web site that a normal user would see. To change the webserver group setting, edit the `webserver_group` line in the [iwserver] section of iw.cfg. If iw.cfg does not contain this line, add it as shown below:

```
webserver_group=TeamSite Web Preview
```

If this line does not exist, the group “Everyone” will be used.

Web Daemon

To set Web daemon defaults, edit the [iwwebd] values in the iw.cfg file:

```
[iwwebd]
host=hostname.domain
http_port=80
https_port=443
```

```
default_protocol=http
```

The `default_protocol` setting is used by the following scripts when TeamSite generates URLs:

- `iwsend_servlet_mail.ipl` script—uses it to embed URLs into the email messages it sends
- `<iwov_webdesk_url>` presentation template tag—uses it when generating hyperlinks to the TeamSite server (see the *TeamSite Templating Developer's Guide* for more information)

Servlet Engine

By default, the servlet port is set to 8080. To change this setting, edit the `servlet_port` value in the `[teamsite_servlet_ui]` section of the `iw.cfg` file.

```
[teamsite_servlet_ui]
servlet_port=8080
```

Main Branch Settings

The following settings apply only to the main branch in TeamSite, not to any of its sub-branches. Because the main branch is not ordinarily used for development, these settings may not apply to your TeamSite configuration. However, if you have a special need to change the locking model, owner, or group of the main branch, you can use the following settings.

Locking Model

TeamSite allows you to specify the locking model of each branch at the time that it is created. However, the main branch is created automatically when TeamSite is installed, or when a new backing store is created. You can specify which locking model to use for the main branch of a new backing store by editing the `main_lock_model` line in the `[iwserver]` section of `iw.cfg` (for a detailed explanation of locking models, refer to the *TeamSite User's Guide*). When TeamSite is first installed, it uses the default option of Submit locking for the main branch. The type of locking a branch uses cannot be changed after the branch has been created. However, if you edit the `main_lock_model` line and then create a new backing store, the new settings will take effect on the new backing store. For information about creating a new backing store, see page 232.

```
main_lock_model=locking_model
```

where *locking_model* is one of `submit_lock`, `optional_write_lock`, or `mandatory_write_lock` (or, more simply, `s`, `o`, or `m`). Submit locking is the default option for the main branch. Optional and mandatory write locking may significantly reduce system performance.

Owner and Group

You can specify the owner and group of the main branch of a new backing store by editing the `main_owner` and `main_group` lines in the `[iwserver]` section of `iw.cfg`. When TeamSite is first installed, it uses the default option of Administrator for main branch ownership. To change this setting on an existing main branch, you must use the Windows NT File Properties to take ownership, or the command-line tool `iwchgrp` to change the group of the root directory of the main branch. However, if you edit the `main_owner` and `main_group` lines and then create a new backing store, the new settings will take effect on the new backing store. For information about creating a new backing store, see page 232.

```
main_owner=Administrator
main_group=Administrators
```

Locked File Submission

You can configure TeamSite to allow only the owner or creator of the lock to submit a locked file to the staging area (as opposed to allowing any member of the workarea where the file is locked). To configure this option, add the following line to the `[iwserver]` section of `iw.cfg`:

```
only_lock_owner_creator_submits=yes
```

Submit and Update Logs

You can configure the number of events that are contained in the Submit and Update logs for a workarea. To change this number, remove the comment (`#`) symbol from the `event_log_size` line in the `[iwserver]` section of `iw.cfg` and edit the line to specify the number of events you want to record. If this line does not appear in `iw.cfg`, add it as shown below. For example, with this setting, the Submit and Update logs will contain the 64 most recent Submit or Get Latest operations (as opposed to the 64 most recent files that were submitted or updated).

```
event_log_size=64
```


You can also configure whether or not you want all the files contained in new or deleted directories to be listed individually in the Submit and Update logs. To configure this option, remove the comment (#) marks from the `full_submitlog` and `full_updatelog` lines in the `[iwserver]` section of `iw.cfg` and edit the lines to specify `yes` to show all the files that were contained within the directory that was added or deleted or `no` to show only the directory names. If these lines do not appear in `iw.cfg`, add them as shown below.

```
full_submitlog=no  
full_updatelog=no
```

Branch and Workarea Security

Branch and workarea security determines whether or not a user can see the names of branches and workareas he does not have access to. If a user does not have read access to a branch or workarea, and branch and workarea security is turned off, he will be able to see the name of the branch or workarea, but it will not be linked, and `[N/A]` will appear next to it. However, you can configure TeamSite to not even show the names of branches and workareas in the TeamSite GUI if the user does not have read permissions. To set this option, remove the comment (#) marks from the `branch_security` and `workarea_security` lines in the `[iwserver]` section of `iw.cfg` and edit the lines to specify `off` to show all branch and workarea names or `on` to show only the branch and workarea names for which the user has read access.

If these lines do not appear in `iw.cfg`, add them as shown below.

```
branch_security=on  
workarea_security=on
```

Domains to Use for Group Authentication

To configure which domains to use for group authentication, configure the `domain_list` line in the `[iwserver]` section of `iw.cfg`. This setting can be used to reduce the startup time for TeamSite, by limiting the number of domains it tries to authenticate against.

To configure this option:

1. If a comment symbol (#) is present at the beginning of the `domain_list` line in the `[iwserver]` section of `iw.cfg`, remove it. Do not confuse this line with the `domain_list` line in the `[iwcgi]` section of `iw.cfg`. If `iw.cfg` does not contain this line, add it as shown below.

2. Edit the line to read:

```
domain_list=domain1, domain2, domain3
```

You can include any number of domains in this list.

Logging Users and Groups

TeamSite can be enabled to record in its log files every authenticated user and all groups with which each user is associated. To activate this feature, add the following line to the `[iwserver]` section of `iw.cfg`:

```
show_user_list=true
```

Once this feature is activated, each time TeamSite is restarted, it will log all users in the roles files and their associated groups in `iw-home\local\logs\iwtrace.log`. Log files will be in the following format:

```
user: username [associated groups]
```

Note: Be careful when using this feature. If it is left on for too long, your log files will grow extremely large.

File Locations

The `[locations]` section of `iw.cfg` may be used to change the locations of various TeamSite files and directories. To change the location of one of the following files or directories, remove the comment (#) marks from its line and edit the line to point to the new location (ensure that the `[locations]` line is not also commented out). After restarting, TeamSite looks for the specified file or directory in the new location.

If you change the location of `iwmount`, you will need to edit its webserver alias to point to the new location. In addition, Registry entries or any existing files in `iw-home\etc\defaultiw*` take precedence over these settings.

If you change the location of one of the logs, and no file of the specified name is present in the new location, a new file will be created.

If `iw.cfg` does not contain these lines, add the ones you want to configure as shown.

```
[locations]
iwbin=C:\Program Files\Interwoven\TeamSite\bin
iwmount=Y:
iwcgimount=Y:
iwroles=C:\Program Files\Interwoven\TeamSite\local\config\roles
iwstore=C:\iw-store
iwsubmitconfig=C:\Program Files\Interwoven\TeamSite\local\config\submit.cfg
iwauprivate=C:\Program Files\Interwoven\TeamSite\local\config\autopprivate.cfg
iwlogs=C:\Program Files\Interwoven\TeamSite\local\logs
iwconfigs=C:\Program Files\Interwoven\TeamSite\local\config
iweventlog=C:\Program Files\Interwoven\TeamSite\local\logs\iwevents.log
iwtracelog=C:\Program Files\Interwoven\TeamSite\local\logs\iwtrace.log
iwdeploylog=C:\Program Files\Interwoven\TeamSite\local\logs\iwdeploy.log
launchpad=C:\Program Files\Interwoven\TeamSite\local\config\launchpad.cfg
```

where:

<code>iwbin</code>	Specifies the location of TeamSite binaries (normally <code>iw-home\bin</code>).
<code>iwmount</code>	Specifies the location of the TeamSite mount point.
<code>iwcgimount</code>	Specifies the location of the TeamSite mount point.
<code>iwroles</code>	Specifies the directory containing the TeamSite roles files.
<code>iwstore</code>	Specifies the location of the TeamSite backing store (this setting can be overridden by the Registry key).
<code>iwsubmitconfig</code>	Specifies the location of the Submit Filtering configuration file.
<code>iwauprivate</code>	Specifies the location of the Autoprivate configuration file.
<code>iwlogs</code>	Specifies the directory containing TeamSite logs.
<code>iwconfigs</code>	Specifies the default configuration file directory.

<code>iweventlog</code>	Specifies the location of the TeamSite event log.
<code>iwtracelog</code>	Specifies the location of the TeamSite trace log.
<code>iwdeploylog</code>	Specifies the location of the deployment log.
<code>launchpad</code>	Specifies the location of the LaunchPad autoconfiguration file (default is <code>C:\Program Files\Interwoven\TeamSite\local\config\launchpad.cfg</code>).

Autoprivate

TeamSite's Autoprivate feature allows you to prevent certain file types and directories, such as temporary files and Macintosh resource forks, from being submitted to the staging area or copied during a **Copy To** operation. File types specified in the Autoprivate configuration files automatically get marked Private. For more information about Private files, see the *TeamSite User's Guide*.

Note: Changes to Autoprivate only apply to files or directories that are created or renamed after the changes are made. Changes do not apply to existing files.

To turn on Autoprivate, create a file named `autoprivate.cfg` in your `iw-home\local\config\directory`. The Autoprivate file consists of two sections:

- files (or directories) matched by pattern
- files (or directories) matched by name

Each section is set off by parentheses on their own lines, and the file begins with a " (" (open parenthesis) on its own line and ends with a ") " (close parenthesis) on its own line.

Individual entries in the first section are in the following format:

```
((filenamepattern)(#_characters_to_match_at_beginning)
(#_characters_to_match_at_end))
```

where both `#_characters_to_match_at_beginning` and `#_characters_to_match_at_end` are in hexadecimal.

For example, to have Autoprivate detect any file or directory that ends with `.frk`, add the following entry:

```
((x.frk)(0)(4))
```

meaning to match zero characters at the beginning of the name and the four characters `(.frk)` specified at the end of the name.

To set Autoprivate to detect any filename that ends in `.backup.fm`, add the following entry:

```
((x.backup.fm)(0)(a))
```

where `0` specifies not to match any characters at the beginning, and `a` (hexidecimal 10) specifies to match ten characters at the end of the filename.

Entries in the second section specify exact filename matches, set off by double parentheses. These filename matches apply across all directories in all workareas on the TeamSite server. For example, if `autoprivate.cfg` includes:

```
((test))
```

then all files and directories named `test` that are created after this line is added, in all directories in all workareas in TeamSite, will be marked private.

The `autoprivate.cfg` file recognizes the following six special characters: `() [] #` and a space (spacebar). If your file names contain any of these characters, you must encode these values when specifying them as a pattern. For example, to have Autoprivate detect a file name that includes spaces, encode the spaces with a `\20`, for example, to match “Network Trash Folder”:

```
((Network\20Trash\20Folder))
```

Encodings are represented as `\xx` where `xx` is the hex value of the corresponding ASCII character. The following table shows the mappings for the six special characters.

Special Character	Autoprivate Encoding
#	\23

Special Character	Autoprivate Encoding
[\5b
]	\5d
(\28
)	\29
space (spacebar)	\20

Encoding examples:

<code>((\23x\23)(1)(1))</code>	matches file names of the form: <code>#*#</code>
<code>((\23bbaax)(2)(0))</code>	matches: <code>#b*</code>
<code>((\28ab\29)(2)(2)) # (ab)</code>	matches the file name: <code>(ab)</code> , the <code>#(ab)</code> is a comment
<code>((a\5b\5db)(2)(2))</code>	matches: <code>a[]b</code>

The following sample `autoprivate.cfg` file includes a few common entries:

```
(
(
((x.o)(0)(2))
((x.a)(0)(2))
((x~)(0)(1))
((.nfsXXX)(4)(0))
((x.bak)(0)(4))
((x.tmp)(0)(4))
)
(
((network\20trash\20folder))
((Network\20Trash\20Folder))
((.HSAncillary))
((.HSResource))
((.hsancillary))
((.hsresource))
((.tnatr:intf))
((.tnatr:reso-fork))
((resource.frk))
((trash))
)
```

)

For changes to `autopriivate.cfg` to take effect, restart the TeamSite server or use the `iwreset` command-line tool.

New File Templates

If you are using templates with TeamSite's New File feature, you can configure which templates are to be used in various parts of the Web site. These settings are controlled through the templating configuration file, `iw-home\local\config\iwtemplates.cfg`.

This file governs which templates are accessible from which directories and branches. To configure which directories a template can be used in, add a line to this file. Only non-TeamSite Templating access is configured in this file (for example, HTML templates or Microsoft Word templates). To configure TeamSite Templating, consult the TeamSite Templating manual.

Syntax

`iwtemplates.cfg` uses the following format:

```
templates
{
  backing_store/main/branch
  {
    template_type
    {
      template_identifier
      {
        template=regular-expression
      }
    }
  }
}
```

where:

- `backing_store/main/branch` specifies the vpath to a branch and backing store in a MultiStore environment. If you are using a single backing store, `branch` specifies the vpath to a branch.
- `template_type` specifies the type of template to be configured in that section.



- *template_identifier* is the individual template identifier that will appear in the New File GUI.
- *template=regular-expression* specifies the full path of each template (rooted in a workarea) and uses case-insensitive regular expression matching to determine which directories this template may be accessed from.

You can have any number of *branch* sections, each of which can have any number of *template_type* sections. The *template_type* sections can have any number of *template_identifier* sections, each of which contains one *template* line.

Each *template* line has the following format:

```
template=regular-expression
```

or

```
template={regular-expression1, regular-expression2,  
regular-expression3...}
```

The left-hand side of each line specifies the template, and the right-hand side contains the case-insensitive regular expression that determines where this template may be used.

Here are some common examples:

To permit the template `global.html` to be used across the entire Web site on a branch, you would add this *template* line to a branch section in `iwtemplates.cfg`:

```
/templates_dir/global.html='/'
```

To permit the template `subdir.html` to be used in any directory path that contains a subdirectory named `subdir`, for example, `/htdocs/subdir/company` or `/htdocs/products/subdir`, you would add this line:

```
/templates_dir/subdir.html='subdir'
```

To permit the template `company.html` to be used in `/htdocs/company` and all of its subdirectories, you would add this line:


```
/templates_dir/company.html='^/htdocs/company/ '
```

To permit the template `products.html` to be used in the `/htdocs/products/` directory but not its subdirectories, you would add this line:

```
/templates_dir/products.html='^/htdocs/products/$'
```

You can also specify multiple patterns to be matched. If you specify multiple patterns, enclose the right-hand side of the line in curly brackets and separate the individual patterns by commas. For example:

```
/templates_dir/products.html={'^/htdocs/products/', 'subdir'}
```

would permit the `products.html` template to be used in `/htdocs/company` and all of its subdirectories, and in any directory path that contains a subdirectory named `subdir`.

Launching Files Through iwproxy

This option enables in-context QA and consistent views of TeamSite workareas. By default, this option is turned on. However, if your Web site must support SSL (Netscape only), you will need to turn this option off and install TeamSite's redirector module, `iwproxy_isapi.dll` or `iwproxy_nsapi.dll`. To install and configure the redirector module, see "Installing the Redirector Module for IIS" on page 48 (for IIS web servers) or "Installing the Redirector Module for NES and iPlanet" on page 47 (for Netscape web servers).

To turn this option off:

1. If a comment symbol (#) is present at the beginning of the `use_iwproxy` line of `iw.cfg`, remove it. If `iw.cfg` does not contain this line, add it as shown below.
2. Edit the line to read:

```
use_iwproxy=no
```

Configuring the TeamSite Server Locale

The `iw.cfg` file now contains a `server_locale` entry in the `[iwserver]` section. The entry specifies the locale in which current execution of the TeamSite server (`iwserver`) runs.

For example:

```
[iwserver]
:
server_locale=English_UnitedStates.MS1252@Binary;
```

This setting is automatically written to the `iw.cfg` file when `iwserver` is started. The system locale is determined by reading the System Locale setting in the **Regional Settings** control panel. Once the `server_locale` setting exists in the `iw.cfg` file, it is used to determine the TeamSite server's system locale at every invocation of `iwserver`. If this setting is not present, `iwserver` determines its locale from the System Locale setting in the **Regional Settings** control panel.

Note: While this setting can be user-modified, it is designed to serve as reference as to the locale in which `iwserver` is currently running. If you have a situation where you want to force `iwserver` to run in a particular locale (independent of the System Locale setting) you can manually set the `server_locale` field.

The locale in which the server operates (as defined by the `server_locale` entry), effectively determines the locale of the TeamSite IFS. For example, if `iwserver` runs under the `Japanese_Japan.Shift_JIS@Binary` locale, all file and directory names are encoded in `Shift_JIS` encoding.

The `server_locale` setting in the `iw.cfg` file can contain any of the locales listed in the following table (note that these settings are Interwoven naming conventions—the operating system locales to which they map are also contained in the table):

iw.cfg server_locale Setting	Windows NT and Windows 2000 Locale
<code>Japanese_Japan.MS932@Binary</code>	Japanese NT and Japanese 2000
<code>German_Germany.MS1252@Default</code>	German 2000
<code>French_France.MS1252@Default</code>	French 2000
<code>English_UnitedStates.MS1252@Binary</code>	U.S. English NT and U.S. English 2000

Configuring Server Performance

Permitting Read-only Operations During Backing Store Freezes

In the process of freezing the TeamSite backing store, TeamSite makes sure that all changes are saved to the backing store from memory. After large operations, such as deleting workareas or editions, updating a workarea, or submitting a large number of files, saving changes to the backing store can take a significant amount of time, during which no other operation is permitted, to minimize the time it takes to save all changes.

This configuration file setting allows read-only operations, such as file system browsing, or navigating through the GUI, to be periodically served during the saving phase of the freeze, at the interval (in seconds) specified by the `flushtimeslice` setting in `iw.cfg`. Giving this setting a smaller value permits better responsiveness, at the expense of causing TeamSite to take longer to save changes. A value of 1 will be quite responsive, and a value of 30 will be very unresponsive. To set this value, edit the following line in the `[iwserver]` section of `iw.cfg`:

```
flushtimeslice=value
```

For example:

```
flushtimeslice=8
```

Cache Size

To set TeamSite's cache size, edit the `cachesize` line in the `[iwserver]` section of `iw.cfg`. If a comment symbol (`#`) is present at the beginning of this line, remove it. If this line does not appear in your `iw.cfg` file, add it as shown below. The initial cache size setting should be approximately three times the number of files and directories on the largest branch.

For example, if the largest branch contains 15,000 files and directories, you should set cache size to 45000 as follows:

```
cachesize=45000
```

Minimum cache size is 1000; maximum is 500000 (five hundred thousand). Each cache line takes a maximum of 1KB of physical memory. Recommended physical memory is cache size

times 1KB plus an additional 25% as a safety margin. In the example shown below, physical memory would be $(45,000 * 1KB) + 11MB = 56MB$. If you encounter a great deal of memory swapping, you should either reduce the cache size or install more memory.

You must restart the TeamSite server for these changes to take effect.

RPC Threadcount

The RPC threadcount setting determines how many simultaneous requests TeamSite can handle from users via the GUI or command-line tools. These requests are very short-lived, so that threads are quickly freed for other users. If all threads are currently being used, TeamSite starts to serialize requests. This setting should not be altered.

```
rpc_threadcount=64
```

File System Threadcount

The file system threadcount should be set to approximately the number of CPUs on the TeamSite server. To change the file system threadcount, edit the `fs_threadcount` line in the `[iwserver]` section of `iw.cfg`. If a comment symbol (`#`) is present at the beginning of this line, remove it. If this line does not appear in your `iw.cfg` file, add it as shown below.

```
fs_threadcount=2
```

You must restart the TeamSite server for this change to take effect.

Filesystem Active Area Cache

The file system active area cache should be set to approximately the number of users who are expected to be using TeamSite concurrently. Note that this is the number of users who are using TeamSite at one time, not the total number of TeamSite users. If this value is too large, it will significantly impact memory usage.

To set the file system active area cache, edit the `fs_active_area_cache` line in the `[iwserver]` section of `iw.cfg`. If a comment symbol (`#`) is present at the beginning of this line, remove it. If this line does not appear in your `iw.cfg` file, add it as follows:

```
fs_active_area_cache=8
```

You must restart the TeamSite server for this change to take effect.

Throughput Monitors

Throughput monitors can be used in conjunction with the `iwstat` command-line tool to monitor system status and performance. To turn on throughput monitors, remove the comment marks (`#`) from the beginning of the lines for the throughput monitors you want to use in the `[iwserver]` section of `iw.cfg`. By default, there are throughput monitors that return system statistics over the previous minute, fifteen minutes, hour, 8 hours, 24 hours, and for the entire time that the system has been running. There are also two throughput monitors that you can configure with any time interval.

```
thruputmonitoring=on
thruputmonitor1=1      # 1 minute
thruputmonitor2=15     # 15 minutes
thruputmonitor3=60     # 1 hour
thruputmonitor4=480    # 8 hours
thruputmonitor5=1440   # 24 hours
thruputmonitor6=-1     # forever
thruputmonitor7
thruputmonitor8
```

Detecting Low Disk Space

TeamSite is configured to freeze the backing store when it detects that free disk space is low. The backing store remains frozen until sufficient disk space is restored, at which point the server returns to its normal running state. This feature helps prevent possible corruption of the backing store. While the backing store is frozen, users cannot write to the TeamSite backing store. Users can still perform read-only operations. The CLT `iwfreeze` can be used to manually freeze the backing store.

The lines shown below in the `[iwserver]` section of `iw.cfg` control the behavior of disk low detection.

The `disklow_mbytes` line gives the server a freeze threshold in MB (the default is 50). The `disklowpercent` line sets the percent of free disk space that is considered “low” (the default is 10). The TeamSite server does not allow `disklowpercent` to go below 2%. If the server detects a low-disk state based on the threshold set in `iw.cfg`, it does not allow you to manually unfreeze the backing store via the `iwfreeze` command. To change these settings, edit these lines as shown below.

```
disklow_mbytes=20
disklowpercent=15
```

Submit Filtering

The TeamSite server allows you to automatically change file attributes, such as owner, group, and ACLs, at the time that you submit a file. This option allows you to automate the task of specifying the permissions that each file will have in the deployed Web site. The submit filter performs the specified operation on files immediately before they are submitted, so that changes are made to the files in the workarea, which are then submitted.

On startup, the TeamSite server reads a configuration file named `submit.cfg` in the `iw-home\local\config\` directory (unless the location of this file is otherwise specified in the `[locations]` section of `iw.cfg`). The `submit.cfg` file contains rules to match file and workarea patterns to specific actions to perform when files and directories are submitted.

It has the following format:

```
case-sensitive = [yes|no]
rules
{
    workarea1_pattern
    {
        file_pattern1 { action1 action2 ... }
        file_pattern2 { action3 action4 ... }
        ...
    }
}
```

```

    }
    workarea2_pattern
    {
        file_pattern3 { action5 action6 ... }
        file_pattern4 { action7 action8 ... }
        ...
    }
    ...
}

```

The case-sensitive statement specifies whether or not the rules matching should ignore the case of the path names. If case-sensitive is not specified, the value is assumed to be `no`.

workarea pattern is used to match a workarea to the set of file rules to apply when a submit is done from the workarea. Each pattern can only be specified once, and the first match is used. The syntax of the pattern is `regex(5)` (extended syntax). For more information on regular expressions, consult a reference manual such as *Mastering Regular Expressions*, by Jeffrey Friedl.

The match is done against the path name of the workarea, starting with `/default/main`.

file pattern is used to match a file or directory to the set of actions to perform on it when it is submitted. Each file or directory pattern can only be specified once, and the first match is used. The syntax of the pattern is `regex(5)` (extended syntax).

The match is done against the path name of the file or directory relative to the workarea.

action is one of

`owner` = *name* (changes the owner of the file or directory)
`group` = *name* (changes the group of the file or directory)
`setaccess` = *ACL* (replaces the access control list for the file or directory)
`changeaccess` = *ACL* (modifies the access control list so that the specified users have only the specified rights)

name is one of

username
groupname



domainname\username
domainname\groupname

ACL stands for Access Control List. An ACL contains one or more Access Control Entries (ACE), as follows:

name:perms (a single ACE, to specify a single user or group)
{ *name:perms*, *name:perms*, ... } (multiple ACEs, to specify multiple users or groups)

where *perms* specifies the permissions granted to the specified user or group and is either any sequence made of the characters:

R (read)
W (write)
X (execute)
D (delete)
P (change permissions)
O (take ownership)

or else one of the preset combinations:

ALL (RWXDPO)
NONE (none)
READ (RX)
WRITE (W)
CHANGE (RWXD)

For example:

```
setaccess={ andre:ALL, marketing\pat:RX }
```

would remove the existing ACL and grant andre full access and pat (in the marketing domain) read and execute access to the specified files.

```
changeaccess={ chris:CHANGE, everyone:RX }
```


would remove any existing ACEs for the user `chris` and the group `everyone`, and grant `chris` change access and the group `everyone` read access to the specified files. Any other existing ACEs would remain unchanged.

When you submit files or directories:

1. The server determines what files and directories have actually changed and need to be submitted. It also verifies that none of them are in conflict with the staging area or locked in other workareas.
2. The path name of the workarea from which the submit is being done is matched against the workarea patterns from the configuration file.
3. If the workarea matches one of the workarea patterns, then, for each file and directory that needs to be submitted (as determined in step 1), the file's path name is matched against the file patterns in the matching workarea's section.
4. If a match is found, then the server performs the specified set of actions to the file or directory in the workarea.
5. The server submits the transformed files and directories to the staging area.

Example

This is a sample `submit.cfg` file:

```
CASE-SENSITIVE = NO
```

```
RULES
```

```
{
.          # any workarea
{
/a/b/.*\.html$ # files ending in .html under /a/b
{
owner=DOMAIN\andre
group="DOMAIN\web editors"
setaccess = {everyone:Read, domain\andre:ALL}
}
[^/]$      # all other files
{
```



```
    group="DOMAIN\\web editors"  
    setaccess = {users:rx, "domain\\webeditors:change"}  
  }  
/$          # all directories  
{  
  group="DOMAIN\\web editors"  
  setaccess = {everyone:rw/rx, "domain\\webeditors:change"}  
}  
}
```

Notes

Only the first match is applied. That is, the first match wins if multiple rules match the file or directory. `submit.cfg` should be ordered so that the most specific workarea patterns are closer to the top and the most specific file patterns are earlier in each section. You may need to duplicate some actions for them to apply to multiple rules.

For purposes of matching, the path name of a directory must end in a slash ("/").

Single or double quotes around patterns are optional, but they must be used around workarea and file patterns that contain white space or other special characters, like #, {, }, =, or ,. Backslashes (\) are special characters when used within patterns surrounded by quotes; a backslash followed by any character is replaced by the single character. For example, to embed a single quote, double quote, or backslash in a pattern, precede the character with a backslash (\', \", or \\). Backslashes are not special characters in patterns that are not quoted.

For example, the following three specifications are equivalent:

```
owner = DOMAIN\andre  
owner = 'DOMAIN\\andre'  
owner = "DOMAIN\\andre"
```

You can use backslashes (\) or forward slashes (/) as the path delimiter in regular expressions, but using forward slashes is much more readable. This is because the backslash is treated as a special character in regex(5) syntax, and it is also treated as a special character by the configuration file parser when the expression is enclosed in quotes or double quotes. Therefore, when an expression using backslashes is contained in quotes or double quotes, the backslashes must be escaped twice, for a total of four backslashes.

For example, the following are equivalent expressions for matching any file whose name ends in `.html` in the `\a\b` directory:

```
^/a/b/.*\..html$
'^/a/b/.*\..html$'
"^/a/b/.*\..html$"
^\\a\\b\\.*\..html$
'^\\\\a\\\\b\\\\.*\..html$'
"^\\\\a\\\\b\\\\.*\..html$"
```

Do not specify duplicate workarea patterns multiple times, duplicate file patterns multiple times within a workarea section, or the same action multiple times within a file rule.

Changes to `submit.cfg` do not take effect until the server is restarted or until you use `iwreset`.

Debugging

The CLT `iwtestcfg` (see *TeamSite Command-Line Tools*) can be used to find out which workarea and file pattern will be applied to a file at the time of submission:

```
>iwtestcfg /default/main/WORKAREA/andre/cgi/test.sh
```

Would return:

```
Matched area pattern "^/default/main/WORKAREA/.*$"
Matched file pattern ".*\..sh$"
Actions to do are:
owner = andre
```

Matched area pattern and Matched file pattern are the case-insensitive regular expressions found in `submit.cfg` that match the workarea and file. Actions to do are the actions (specified in `submit.cfg`) that will be applied to the file.

You can also get debugging information on the submit handling configuration printed to `iwtrace.log`, by adding the following line to the `[server]` section of `iw.cfg`:

```
debug_event_handler=yes
```

This will cause the server to print a configuration map of `submit.cfg` and to print which actions are performed as files are submitted.

Configuring the TeamSite Web Daemon and Proxy Server

About the TeamSite Web Daemon

TeamSite uses a Web daemon, `iwwebd.exe`, to provide SSL support for the TeamSite browser GUI. Remote contributors can use TeamSite securely without having to establish a Virtual Private Network (VPN). This Web daemon also serves up the non-servlet-based parts of the TeamSite GUI.

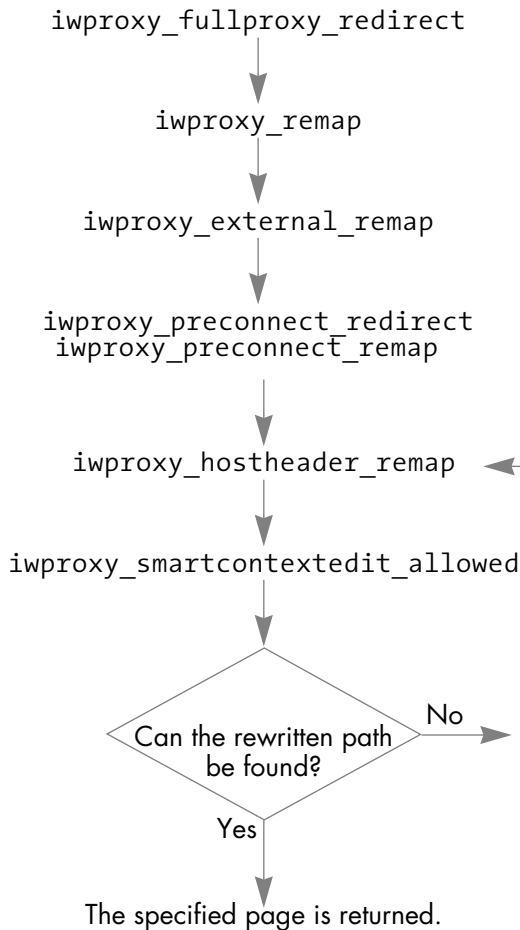
For an illustration of how requests are processed, see “TeamSite Architecture” on page 21.

About the Proxy Server

TeamSite uses a proxy server to perform several important functions:

- Resolve relative and absolute URL names in TeamSite areas in order to present users with a virtualized view of the Web site contained within an area (see page 167).
- Redirect fully-qualified URLs into TeamSite areas (see page 171).
- Redirect browsing in one branch or workarea into another area (see page 175).
- Redirect individual workareas to use different Web servers (see page 178).
- Remap links to external Web servers (see page 179).
- Modify “Host:” headers (see page 180).
- Remap SSI requests (page 181).

Each time a request is made through the TeamSite proxy server, the following sections of `iw.cfg` are processed in the following order. More than one rule may be applied to a request. As a URL gets rewritten by a rule, the rewritten URL is passed to the next section. The first rule that matches in any section prevails; no other rule in that section will be applied.



See *Configuring TeamSite to Redirect Fully-Qualified Paths*. Applies only if the browser's proxy has been set to the TeamSite proxy server.

See *Document Roots*.

See *Configuring External Remappings*. Applies only if none of the preceding rules has matched.

See *Redirecting TeamSite Views to Different Areas*.

See *Host Header Remappings*.

See *Enabling and Disabling SmartContext Editing*.

`iwproxy_failover_remap`

See *Configuring Proxy Failover*. Sends the rewritten path to be re-processed.

Applying Changes to Proxy Configuration

If you change the `iwproxy` mappings in `iw.cfg`, you will need to reset the server with the `iwreset -a` command-line tool to reflect the changes.

Note that `iwreset -a` will not apply changes to the `[iwproxy_remap]` or `[iwproxy_plugin_remap]` sections of `iw.cfg`, if you are using Web server plugins. If you make changes to these sections, and you are using webserver plug-ins, you will need to restart the Web server to apply the changes.

Configuring TeamSite Web Daemon and Proxy Server Operation

The `[iwproxy]` section of `iw.cfg` is used to configure the operation of TeamSite's proxy server. For example:

```
[iwproxy]
iwproxy_port=1080
iwproxy_host=proxy_hostname
customer_webserver_port=81
customer_webserver_host=hostname
```

where:

`iwproxy_port` is the port TeamSite's proxy server will operate on. It should be set to an open port value (1080 is selected by default).

`iwproxy_host` specifies the host where the TeamSite proxy daemon runs. Usually this will be the TeamSite server.

`customer_webserver_port` is the port through which TeamSite's proxy server communicates with the Web server. It must be set to the value of the port used by the Web server. Port 81 is selected by default.

`customer_webserver_host` is the host name of the content Web server. The value must be set to the host name of the Web server that serves the content of your Web sites.

The settings in the `[iwproxy]` section are set during installation, and can be edited when necessary.

Resolving Relative and Absolute Paths

About Relative and Absolute Paths

Relative paths specify file locations relative to the referencing file's directory location. Absolute paths specify file locations relative to the Web site's document root directory. For example, the file whose directory path (rooted in a TeamSite area) is:

```
/main/index.html
```

might contain a link to the file

```
/images/banner.gif
```

This link can be specified as either a relative or an absolute path.

If the link were specified as a relative path, it would look like:

```
../images/banner.gif
```

If the link were specified as an absolute path, it would begin with a / and look like:

```
/images/banner.gif
```

Note: The proxy server does not allow you to remap the document root directory for backing store branches other than the default store.

Resolving Relative and Absolute Paths

Links in HTML documents are often specified with relative or absolute path names. For example, in a link to an image, the file name might appear as:

```
/images/pic.gif
```

On a typical Web server, this link reference would be mapped by the Web server to its document root, for example:

```
/images/pic.gif ==> D:\inetpub\wwwroot\images\pic.gif
```



All users attempting to access the file using the absolute path name will be mapped to the same file location on the Web server.

However, TeamSite supports a system of private workareas, giving each user access to the Web site's files from within their own personal, virtual version of the Web site. TeamSite uses a proxy server to manage mapping of files to workareas with relative and absolute path references. Going back to our example, the TeamSite proxy server allows each user attempting to access `/images/pic.gif` from within TeamSite to be mapped to the copy of `pic.gif` in his own workarea. The redirected mapping would look like:

```
/images/pic.gif ==>  
Y:\default\main\branchpath\WORKAREA\workareaname\images\pic.gif
```

Document Roots

TeamSite maps the initial Web server directory structure (*document root*) of workareas to the top level of the workarea directory by default. You may, however, want to move the document root, or group types of files together for improved clarity, convenience, or efficiency. On a branch-by-branch basis, the TeamSite proxy server allows you to remap the document root anywhere within the workarea directory. It also allows you to define mappings directly to sub-directories within workareas.

Path mappings are defined by including sections within the TeamSite main configuration file (`iw.cfg`).

To configure document roots for individual branches:

1. For each branch that you want to configure, add a line to the `[iwproxy_remap]` section of `iw.cfg`, of the form:

```
configsectionname=vpath
```

where *vpath* is the vpath to the branch you are configuring, and *configsectionname* is the name of the section of the configuration file that will define the branch remappings.

2. For each line that you added to `[iwproxy_remap]`, create a section in `iw.cfg` named `[configsectionname]`. Add a line to this section that defines the document root:

```
_docroot=dirpath
```

where *dirpath* is a directory path rooted in a workarea.

You can also add lines that bypass the document root, of the format:

```
path=path
```

For example, you might add the following lines to `[iwproxy_remap]`:

```
[iwproxy_remap]
branchrewrite_1=/main
branchrewrite_2=/main/training
```

The first line of the above example tells TeamSite to use the section `[branchrewrite_1]` to set the document root configuration for the `/main` branch. The second line tells TeamSite to use the `[branchrewrite_2]` section to set the document root configuration for the `/main/training` branch.

You would then create two new configuration file sections corresponding to the lines in `[iwproxy_remap]`:

```
[branchrewrite_1]
_docroot=/htdocs
/pictures/=/pictures/
/html/=/html/

[branchrewrite_2]
_docroot=/htdocs
/images/=/images/
```



The first line of both the new sections contains:

```
_docroot=/htdocs
```

This defines a special directive that sets the mapping of the document root. Any requests from workareas on the main branch or the main/training branch to the root level directory (/) will now start at:

```
.../workareaname/htdocs/
```

Thus, the request for file /picture1.gif will now be mapped to:

```
.../workareaname/htdocs/picture1.gif
```

newly defined docroot *file*

The two docroot configuration sections also contain lines similar to the following:

```
/pictures/=/pictures/
```

This line maps file requests directly to the listed directory /pictures/, bypassing the document root defined in the first line. Thus, a request for the file /pictures/people.gif gets mapped to:

```
.../workareaname/pictures/people.gif
```

not:

```
.../workareaname/htdocs/pictures/people.gif
```

TeamSite's proxy server operates using literal string matches and substitutions in path names. To avoid inadvertently rewriting names, always use trailing slashes in your remap definitions (but not your _docroot directories.)

Note: Do not use trailing slashes in your remap definitions for _docroot directories.

Resolving Fully-Qualified URLs

TeamSite's proxy server can also be configured to resolve fully-qualified paths. For example, a link to the main page of a Web site might appear as

```
http://www.name.com
```

If such a link appears in an HTML file in a TeamSite workarea, and you follow that link while performing in-context QA, you will be taken out of the workarea and to the actual referenced Web site.

Therefore, if you use fully-qualified URLs to reference pages within your own Web site, clicking on these links will take you out of the in-context view of the current workarea, staging area, or edition and into your own currently deployed Web site. To solve this problem, TeamSite allows you to configure your proxy server. The proxy server will redirect fully-qualified links within your Web site, then pass them to the regular proxy server to ensure the integrity of the in-context view in a workarea, staging area, or edition.

Note: *Only* configure this setting if your Web site uses fully-qualified URLs that you need to view in-context! This setting requires you to manually configure your browser, so that you will not be able to view the actual Web site without reconfiguring your browser. Also, this slows the TeamSite server by sending every request through the Web daemon and iwproxy.

Configuring TeamSite to Redirect Fully-Qualified URLs

To configure TeamSite to redirect fully-qualified URLs, you must:

- Configure the TeamSite proxy server.
- Set your (client) browser's proxy to the TeamSite Web daemon.

Configuring the TeamSite Proxy Server to Redirect Fully-Qualified URLs

To configure the TeamSite server to redirect fully-qualified URLs, edit the [iwproxy_fullproxy_redirect] section of iw.cfg. This section contains any number of _regex lines. Each line is of the format:

```
_regex=source_regex=dest_ex
```

where *source_regex* is a case-insensitive regular expression specifying a fully-qualified URL that might appear in a page, and *dest_ex* is an expression specifying the path that the link will be redirected to. This expression should always be the path to the file specified in *source_regex*, but rooted in a TeamSite area.

For example:

```
[iwproxy_fullproxy_redirect]  
_regex=http://www(\.example\.com)?/(.*)=/$2
```

redirects links that specify

`http://www.example.com`

in the URL and sends them to the corresponding location in the current TeamSite area.

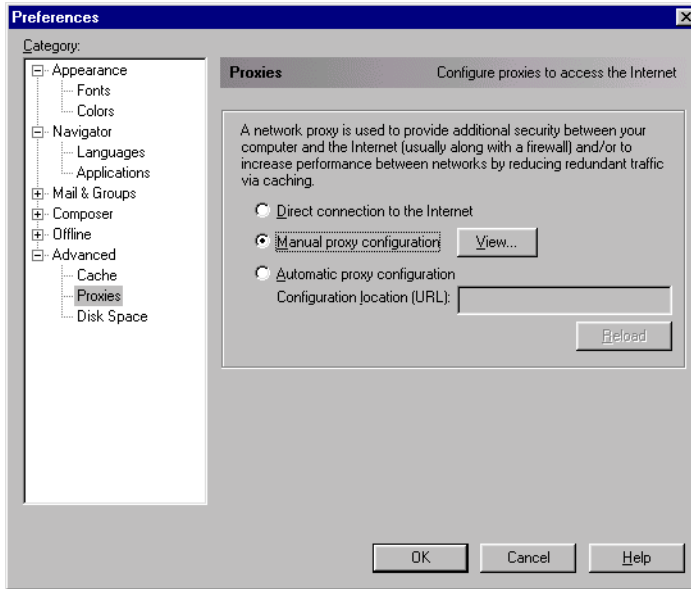
Configuring the Client for Fully-Qualified URL Redirection

If you are using [iwproxy_fullproxy_redirect], you must set up your (client) browsers to go through the TeamSite Web daemon. All requests will then go through iwwebd. If you need to browse one of the live Web sites that the TeamSite Web daemon reroutes requests for, you will need to set your browser to not use the TeamSite Web daemon.

Netscape

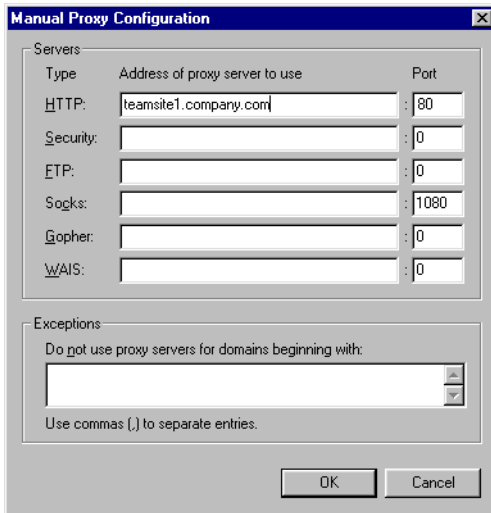
To configure your browser to use the TeamSite Web daemon:

1. In Netscape Navigator, select **Preferences** from the **Edit** menu.
2. The Preferences window will appear. Double-click on the **Advanced** category in the left-hand pane.
3. Select the **Proxies** subcategory in the left-hand pane.



The Netscape Preferences window

4. In the right-hand pane, select **Manual proxy configuration**. Select **View**.
5. The Manual Proxy Configuration window will appear. Type the name of your TeamSite server (for example `teamsite1.example.com`) in the **HTTP** section. Type the `iwwebd_port` specified in the `[iwwebd]` section of `iw.cfg` (for example, 80) in the **Port** section.



The Manual Proxy Configuration window

6. Click **OK**.
7. In the Preferences window, click **OK**.

To configure Netscape to not use the TeamSite proxy server:

1. In Netscape Navigator, select **Preferences** from the **Edit** menu.
2. The Preferences window will appear. Double-click on the **Advanced** category in the left-hand pane.
3. Select the **Proxies** subcategory in the left-hand pane.
4. In the right-hand pane, select **Direct connection to the Internet**.
5. Click **OK**.
6. In the Preferences window, click **OK**.

Internet Explorer

To configure your browser to use the TeamSite proxy server:

1. In Internet Explorer, select **Internet Options** from the **View** menu. The Internet Options window will appear.
2. Select the **Connection** tab.
3. Select the **Access the Internet using a proxy server** checkbox.
4. Type the name of your TeamSite server (e.g. `teamsite1.example.com`) in the **Address** section. Type the `http-port` specified in the `[iwwebd]` section of `iw.cfg` (for example, 80) in the **Port** section.
5. Click **OK**.

To configure Internet Explorer to not use the TeamSite proxy server:

1. In Internet Explorer, select **Internet Options** from the **View** menu. The Internet Options window will appear.
2. Select the **Connection** tab.
3. Deselect the **Access the Internet using a proxy server** checkbox.
4. Click **OK**.

Redirecting TeamSite Views to Different Areas

TeamSite's proxy server allows web teams to work on branches of development that are populated only with the portion of the Web site that they are developing, but still maintain a fully in-context view of the entire Web site by referencing the staging area or a known edition on another branch of development.

This feature is very flexible in that it can be configured on a per-branch or per-workarea basis, and the redirected view can be configured to take the user to any TeamSite area on any branch. Redirection can occur in one of two ways:

1. Through `[iwproxy_preconnect_remap]`, which retains your original area as the current working area and directs files there from another area. In this scenario, docroot is based on the original area's parent branch.
2. Through `[iwproxy_preconnect_redirect]`, which causes the area you redirect into to become the current working area (and that area's parent branch becomes the basis of docroot).

Using `[iwproxy_preconnect_remap]`

To configure TeamSite to redirect workarea views as described in Item 1 above, edit the `[iwproxy_preconnect_remap]` section of `iw.cfg`:

```
[iwproxy_preconnect_remap]
_regex=source_regex=dest_ex
```

where `source_regex` is a case-insensitive regular expression describing the area to be mapped from, and `dest_ex` is an expression describing the area to be mapped to. These areas are most commonly workareas or staging areas, but you can map to and from any workarea, staging area, or edition. You can add any number of `_regex` lines to this section.

For example:

```
_regex=(.*)/branch1/WORKAREA/[^/]+/products/(.*)=$1/branch2/
STAGING/products/$2
```

tells the proxy server to remap the `products` directory of any workarea on any branch named `branch1` to the `products` directory of the staging area on its sister branch, `branch2`.

In the source regular expression, `(.*)` is used to specify an arbitrary path, and `$1` in the destination expression means that it must follow the same path (and therefore `branch1` can be anywhere in the branch structure, but `branch2` is a sister branch of `branch1`). Also in the source regular expression, `[^/]+` is used to specify a single directory level, of any name (which in this case would be the workarea name, and therefore all workareas on `branch1` are specified).

Finally, the source regular expression uses (.*) to specify another arbitrary path, and \$2 in the destination expression tells it to follow the same path.

You can also specify the exact location of the areas you want to remap:

```
_regex=^/
iw-mount/default/main/branch1/WORKAREA/[^/ ]+/products/(.*)=/
iw-mount/default/main/branch2/STAGING/products/$1
```

Or, you can specify an individual workarea to remap:

```
_regex=^/
iw-mount/default/main/dev/branch1/WORKAREA/andre/coolstuff/(.*)=/
iw-mount/default/main/branch2/STAGING/coolstuff/$1
```

The TeamSite proxy server applies the first match it finds, so you can exclude a particular area from a more general rule by creating a separate rule for that area and placing it before the more general rule. For example:

```
_regex=(.*)/branch1/WORKAREA/chris/products/(.*)=$1/branch1/
STAGING/products/$2
_regex=(.*)/branch1/WORKAREA/[^/ ]+/products/(.*)=$1/branch2/
STAGING/products/$2
```

remaps the products directory in all workareas on branch1 except for Chris's to the staging area of branch2.

See “Configuring Proxy Failover” on page 181 for a details about configuration rule precedence.

Using [iwproxy_preconnect_redirect]

To configure TeamSite to redirect workarea views as described in Item 2 above, edit the [iwproxy_preconnect_redirect] section of iw.cfg:

```
[iwproxy_preconnect_redirect]
_regex=source_regex=dest_ex
```

where *source_regex* and *dest_ex* are as described in “Using [iwproxy_preconnect_remap]” on page 176. If you set [iwproxy_preconnect_redirect] and then click on a link defined by an absolute path name, the docroot of that link is based on the branch you redirected into (as opposed to the branch of the area you redirected from, which would be the behavior if you had set [iwproxy_preconnect_remap]). See “Configuring Proxy Failover” on page 181 for a details about configuration rule precedence.

Configuring TeamSite to Use Different Web Servers

You can configure TeamSite to use different Web servers for different workareas or different types of content. For example, Andre might want to make all CGI's in his workarea on branch1 (subject to no constraints whatsoever on the arguments these CGI's or may not take) be served by test1.example.com:1234. This would let Andre test different Web server configurations for his CGI's on branch1 without disturbing anyone else.

To configure TeamSite to use different Web servers, edit the [iwproxy_preconnect_remap] section of iw.cfg:

```
[iwproxy_preconnect_remap]
_regex=source_regex=dest_ex
```

where *source_regex* is a case-insensitive regular expression describing the area and files to be served by the other Web server, and *dest_ex* is an expression describing the area and files on the other Web server. This expression must include the port number.

For this to work properly, the other Web server must have the appropriate TeamSite directory mounts and privileges. The Web server alias used by httpd on port 1234 of test1.example.com must be configured with the TeamSite alias as well (/iw-mount/).

The following example would allow Andre to test all CGIs in his workarea on test1.example.com, as described above:

```
[iwproxy_preconnect_remap]
_regex=^/iw-mount/default/main/branch1/WORKAREA/andre/(.*)\.cgi
(\?.*)?$=http://test1.example.com:1234/iw-mount/default/main/branch1/
WORKAREA/andre/$1.cgi$2
```

Configuring External Remappings

The TeamSite proxy server allows you to define mappings to directories outside of the TeamSite system or on different computers altogether. You can define these mappings through either of the following ways:

- [iwproxy_preconnect_remap]
- [iwproxy_external_remap]

If you use [iwproxy_preconnect_remap], these mappings will follow normal [iwproxy_preconnect_remap] precedence rules. However, [iwproxy_external_remap] mappings apply *only* if no other remapping rule has been applied.

[iwproxy_preconnect_remap]

To configure TeamSite to redirect workarea views to external Web servers, edit the [iwproxy_preconnect_remap] section of iw.cfg:

```
[iwproxy_preconnect_remap]
_regex=source_regex=dest_ex
```

where *source_regex* is a case-insensitive regular expression describing the area to be mapped from, and *dest_ex* is an expression describing the area to be mapped to. These areas are most commonly workareas or staging areas, but you can map to and from any workarea, staging area, or edition.

For example:

```
_regex=(.*)/branch1/WORKAREA/[^/]+/logos/(.*)=http://corporateidserver
.example.com/logos/$2
```

will send all requests for files in the /logos directory in all workareas on branch1 to another server, corporateidserver.example.com.

[iwproxy_external_remap]

You can also use [iwproxy_external_remap] rules for external remappings. This usage is being phased out; use [iwproxy_preconnect_remap] whenever possible.

For example, if all your corporate logo files reside on a separate server, you can use [iwproxy_external_remap] to create a mapping to the directory where they reside:

```
[iwproxy_external_remap]
/logos/=http://corporateidserver.example.com/logos/
```

This remapping sends all requests for /logos/ to a directory on another server, corporateidserver.example.com/logos/. You can also create associations using case-insensitive regular expression mapping.

The [iwproxy_external_remap] section is read after the [iwproxy_remap] section, and it will only be applied if none of the [iwproxy_remap] rules were invoked. For example, if you create a mapping for /logos/ in one of the [branchrewrite] sections, all requests on that branch for files in the /logos/ directory will use that mapping *instead of* the external mapping. Requests on other branches will still be sent to the corporateidserver.

Host Header Remappings

If your Web server manipulates “Host:” headers (for example, virtual domains), you can configure TeamSite to have the same behavior. To configure “Host:” header remapping, edit the [iwproxy_hostheader_remap] section of iw.cfg.

```
[iwproxy_hostheader_remap]
_regex=source_regex=dest_ex
```

where *source_regex* is a case-insensitive regular expression describing the area to be mapped from, and *dest_ex* is an expression describing the new “Host:” header. For example:

```
_regex=^/iw-mount/default/main/branch1/WORKAREA/.*=example.com:1234
```

will change the “Host:” header that the origin server gets from the TeamSite proxy server to read:

```
Host: example.com:1234
```

whenever content in a workarea on `branch1` is accessed.

Configuring SSI Remapping

The TeamSite Web server plug-in supports the ability to both remap and virtualize SSI requests. To enable SSI request virtualization, you must install the necessary redirector module (`iwproxy_nsapi.dll` or `iwproxy_isapi.dll`) in addition to the Web server plug-in.

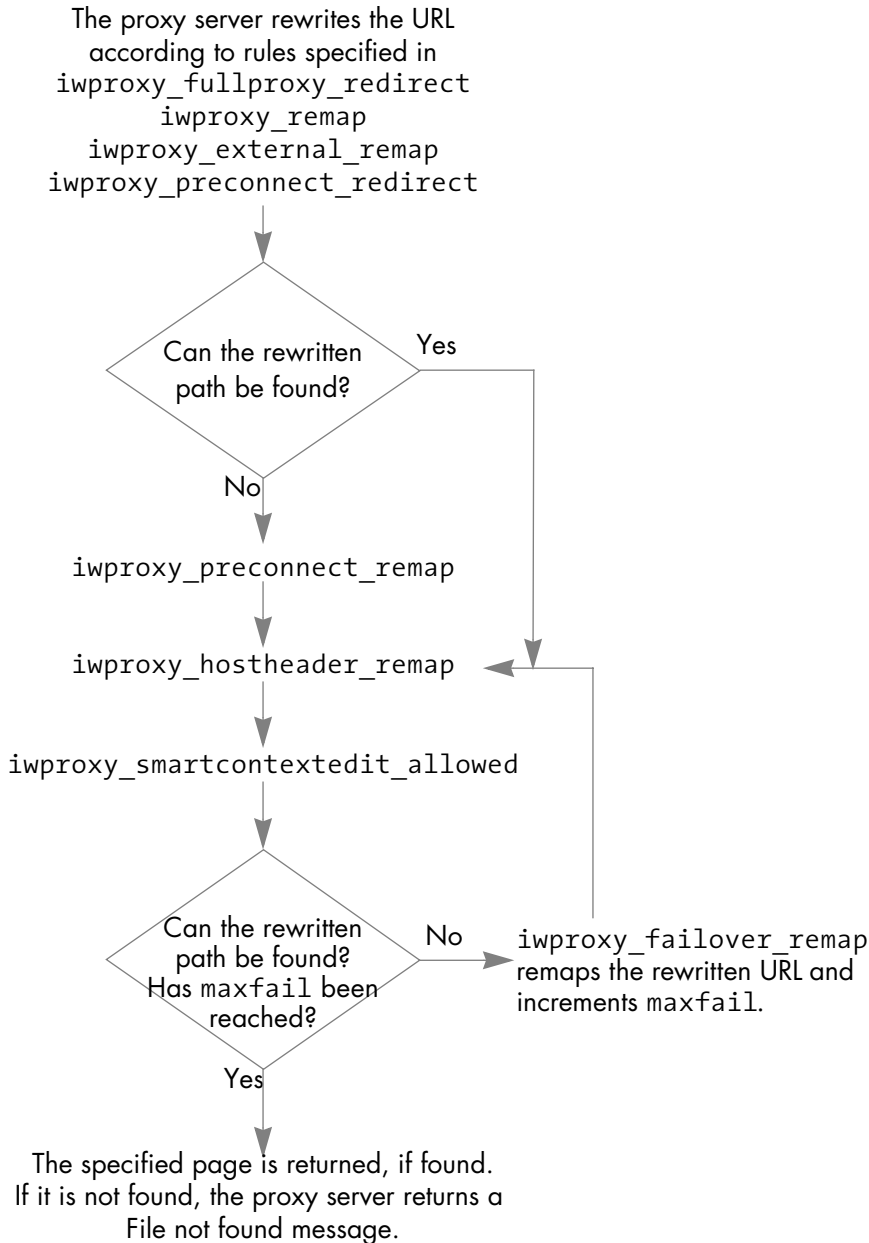
After installing the necessary redirector module as described on page 47, you can configure TeamSite to remap SSI requests by adding or modifying the `[iwproxy_plugin_remap]` section of `iw.cfg`. In the following example, any SSI request containing the string `/forms/` is mapped to `/iw-mount/default/main/Branch2/STAGING/forms` instead of being referred to the root of the user’s workarea:

```
[iwproxy_plugin_remap]
_regex=(.*)/forms/(.*)=/iw-mount/default/main/Branch2/STAGING/forms/$2
```

If you want to debug regular expressions, set the value for `_debug` in the `[iwproxy_plugin_remap]` section to `true`. On NES, debugging information is stored in the Web server error log file. On IIS, this information is stored in `C:\temp\iw_isapi.log`. This log file can grow extremely large over time.

Configuring Proxy Failover

If a requested page does not exist, the `[iwproxy_failover_remap]` section of `iw.cfg` can be used to specify an alternate location. This section allows you to specify both alternate locations and the number of times to process an URL in an attempt to find a valid location. The figure below illustrates the process by which proxy failover remaps URLs.



The `[iwproxy_failover_remap]` section has the following structure:

```
[iwproxy_failover_remap]
_maxfail=#
_regex=source_regex=dest_ex
_regex=source_regex=dest_ex
```

To specify the number of times to try to remap a URL, edit the `_maxfail` line of the `[iwproxy_failover_remap]` section of `iw.cfg`. The default value of this line is `_maxfail=0`, which turns off proxy failover. Note that proxy failover is seldom needed because files are almost always in locations that can be specified via static, case-insensitive regular expressions during configuration. If you need to enable proxy failover, it is recommended that you do not set `_maxfail` to more than 1 or 2 due to the impact on system performance.

To specify expressions to remap, add `_regex` lines to `[iwproxy_failover_remap]`. These lines specify an incoming pattern to match, and an expression that they should be mapped to. The proxy server will take the first match it finds, remap it as specified, then try to locate the page. If it cannot find the new location, it will try to match the remapped expression to a regular expression specified in `[iwproxy_failover_remap]`. This process will continue until a match is found or the number of iterations specified by the `_maxfail` line is reached.

`_regex` lines in the `[iwproxy_failover_remap]` section follow the same syntax as `_regex` lines specified in the `[iwproxy_preconnect_remap]` section of `iw.cfg`, where `source_regex` is a case-insensitive regular expression describing the area to be mapped from, and `dest_ex` is an expression describing the area to be mapped to. For examples of `_regex` syntax, see “Resolving Relative and Absolute Paths” on page 167.

Debugging Your Proxy Server Configuration

If your proxy server does not seem to be configured correctly, use the `iwproxy.exe` CLT’s debug option to list all the translations being made by the proxy server:

```
iwproxy [-d|-x]
```

<code>-d</code>	Debug mode (outputs client & server headers)
<code>-x</code>	Extended (verbose) debug mode (outputs client body text as well)

`iwproxy` will return debug output which you can redirect to a file. Note that `iwproxy`'s debug mode is single-threaded; it therefore slows the TeamSite server down tremendously. Use the debug mode for diagnostic purposes *only*.

One common source of proxy configuration problems is the inclusion of any character or blank space past the end of a branch name in any line in any `[iwproxy*]` section in `iw.cfg`. For example, the following line in the `[iwproxy_remap]` section is illegal because it contains blank spaces and characters after the branch name:

```
[iwproxy-remap]
tag_engspecs=/main/engspecs #This is the engineering spec site
```

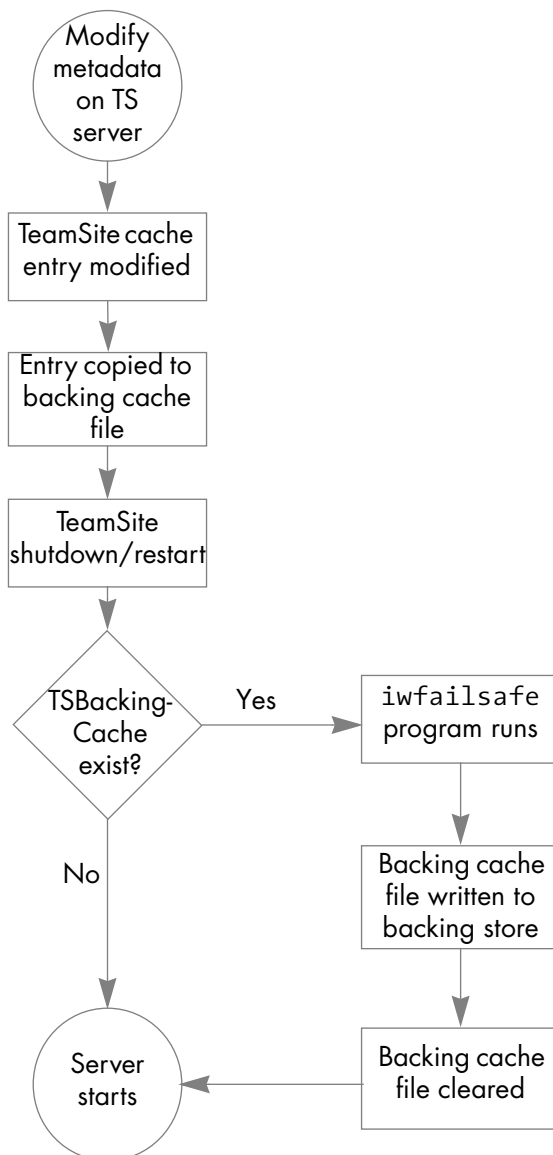
Note: `iwproxy.exe` needs to run as local Administrator, or a user with the following access privileges: Act as Part of Operating System, Log on Locally, Increase Quotas, and Replace a Process Level Token.

TeamSite Embedded Failsafe

The TeamSite Failsafe functionality has been automated to improve the ability to protect your assets against unexpected server outages. Unlike previous versions of TeamSite, there is no need to modify your `iw.cfg` file to benefit from what is now known as *Embedded Failsafe*.

Embedded Failsafe improves reliability by automatically copying TeamSite cache entries to a temporary disk backup file. If the TeamSite server terminates abnormally, these cache entry copies are accessed automatically to restore the backing store when you restart the TeamSite server. This feature significantly reduces the likelihood of metadata inconsistencies caused by abnormal server termination.

The following flowchart shows the processes involved in both normal and abnormal TeamSite shutdowns.



TeamSite Failsafe Process Flow

Configuring Metadata Capture and Search

TeamSite metadata capture lets end users add metadata information to files. After the metadata is deployed to a database via DataDeploy, end users can use TeamSite metadata search to query the database and locate files having specific metadata characteristics.

You must configure TeamSite to enable metadata capture or search; they do not appear by default in the TeamSite GUI. Configuration involves editing a set of configuration files to specify the appearance and behavior of the metadata forms, and then editing the main TeamSite configuration file (`iw.cfg`) to add metadata capture or search to a TeamSite GUI menu. After configuration is complete, end users enter information in either a metadata entry form or a metadata search form. Following data entry, the forms are processed by the metadata capture or metadata search subsystem residing on the TeamSite server. Metadata capture and search exist as separate entities, each accessed via its own TeamSite GUI menu item. Metadata capture can exist without metadata search. However, metadata search requires that you also configure metadata capture.

The rest of this chapter describes how the metadata capture and search subsystems work, and how to configure them. For details about using metadata capture and metadata search, see the *TeamSite User's Guide*.

Metadata Capture

The following sections describe:

- A metadata capture overview.
- The main components that make up metadata capture.
- How to configure metadata capture.

Overview

Metadata capture is a file-specific feature. That is, you must explicitly select the file(s) on which you intend to set metadata. You cannot globally set metadata for an entire area or branch. For example, to set metadata on all files in a workarea, you must select each file in that workarea (by choosing **Select All**, or by clicking the checkbox next to each file, etc.) and then initiate a metadata capture session. See the *TeamSite User's Guide* for more information.

Metadata capture can be initiated in one of two ways:

- Through a job as part of a `<cgitask>` element (see “Initiating Metadata Capture from a Job Specification File” on page 211), or
- From a menu item in the TeamSite GUI. A menu item for metadata capture is not on a TeamSite menu by default; you must add it as described later in this chapter.

Components

No matter how metadata capture is initiated, it relies on four main components:

- The `iw-home\local\config\metadata-rules.cfg` configuration file, which maps vpaths to the data capture rules defined in `datacapture.cfg`.
- The `iw-home\local\config\datacapture.cfg` configuration file, which defines rule sets for capturing data.
- The metadata capture CGI `iwmetadata.cgi`, which interprets data from end users and rules in `datacapture.cfg` and `metadata-rules.cfg`, produces browser graphics and prompts, and acts as an interface with workflow configuration files (if metadata capture is running as part of a job).
- A browser interface for end-user input.

Two configuration files (`metadata-rules.cfg` and `datacapture.cfg`) allow you to configure the following on a per-user or per-vpath basis:

- The metadata item name that is displayed in the metadata entry form.
- The interface through which an end user enters input (for example, a checkbox or a data field).

- The type of data that is acceptable or unacceptable in any given field.
- Whether input is required for any given field.

The following diagram shows how these components work together. Sections following the diagram explain each diagram step and component in detail.

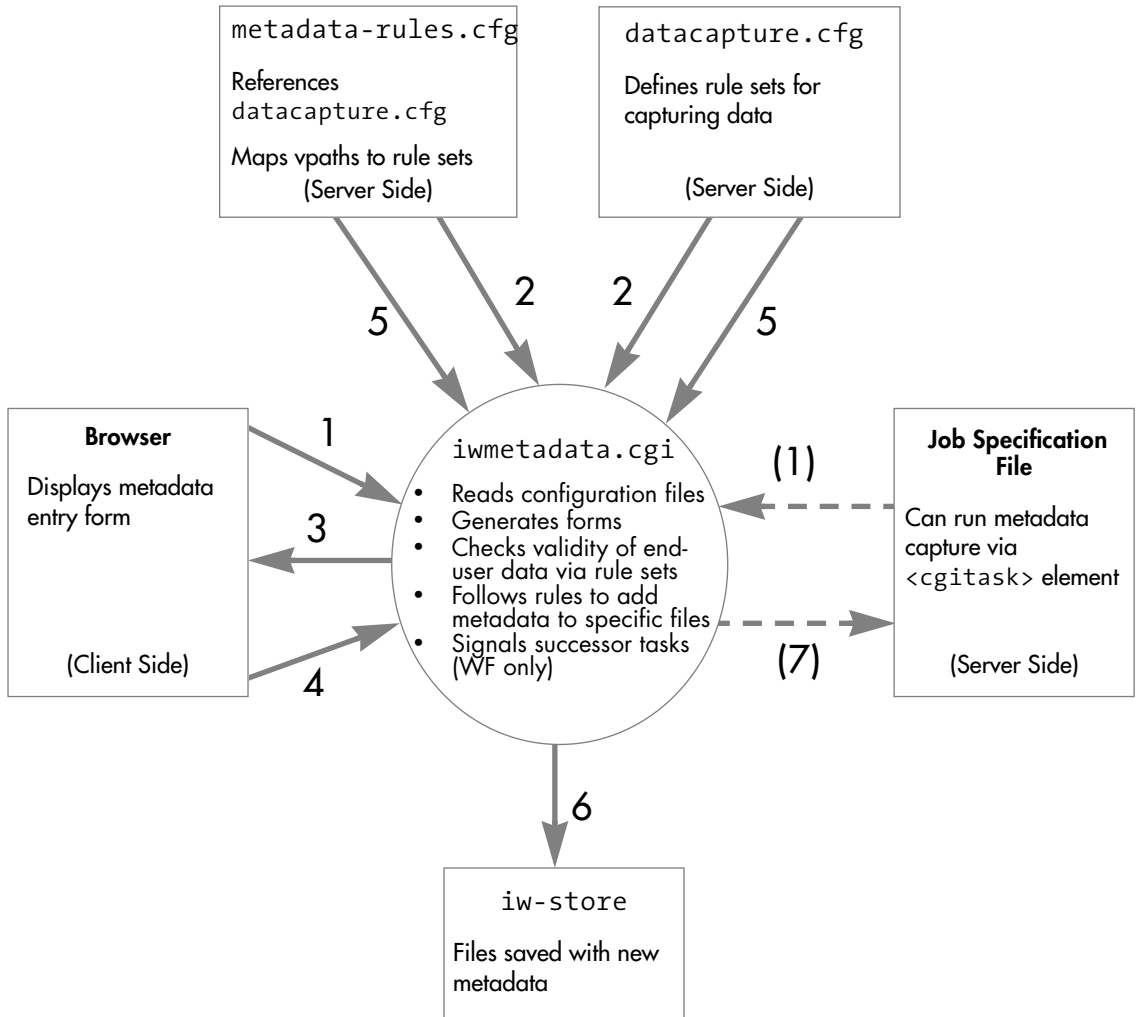


Diagram Key

1. The metadata CGI receives a list containing the names of the files that will have metadata added to them. The list can come from an instantiated job (if metadata capture is initiated from a job) or from the browser (if initiated from the TeamSite GUI).
2. The metadata CGI reads both configuration files (`metadata-rules.cfg` and `datacapture.cfg`) to determine what information it should display in the metadata entry form. It makes this determination on a per-file basis, so that the entry form can contain different prompts and actions for different files.
3. The metadata CGI displays the metadata entry form on the client system via the GUI.
4. An end user fills in data and submits the entry form back to the metadata CGI.
5. The metadata CGI consults the rules in both configuration files to verify the validity of the data entered by the end user. If the data does not meet all necessary criteria, notification is sent to the end user so that data can be re-entered.
6. If the data meets all necessary criteria, the metadata CGI adds the new metadata (in the form of TeamSite extended attributes) to the specified files. The metadata CGI interfaces directly with the backing store to update the files with the new metadata.
7. If metadata capture was initiated from a job, the metadata CGI notifies the workflow subsystem, which starts successor task 0 (zero) as defined in the job specification file.

Configuring Metadata Capture

You must perform three main activities to configure metadata capture:

1. Create a `metadata-rules.cfg` file in `iw-home\local\config` for your site.
2. Create a `datacapture.cfg` file in `iw-home\local\config` for your site.
3. Add a **Set Metadata** item to the TeamSite GUI so that end users can access metadata capture.

The following sections describe these steps in detail.

Configuring metadata-rules.cfg

The `metadata-rules.cfg` file maps vpaths to data capture rules that are defined in `datacapture.cfg`. The `metadata-rules.cfg` file consists of a series of `<cond>` (conditional) elements. A `<cond>` element can contain `<rule>` elements and other `<cond>` elements. Each vpath is run through `metadata-rules.cfg`, resulting in a one-to-many mapping from vpaths to named rules. Whenever a list of `<cond>` elements is found, the first to match the current vpath takes effect, and the rest of the elements in the list are discarded.

When you set up `iw-home\local\config\metadata-rules.cfg` for your site, it is recommended that you copy and edit the example file provided with TeamSite (`iw-home\local\config\metadata-rules.cfg.example`). Use the following DTD and annotated examples as references for your own site-specific configuration.

DTD: metadata-rules.cfg

The `metadata-rules.cfg` file uses the following DTD:

```
<!ELEMENT metadata-rules (cond)*>
<!ELEMENT cond (cond|rule)*>
  <!ATTLIST cond
    vpath-regex CDATA #REQUIRED
  >
<!ELEMENT rule EMPTY>
  <!ATTLIST rule
    name CDATA #REQUIRED
  >
```

Sample metadata-rules.cfg File 1

The following `metadata-rules.cfg` file is distributed with TeamSite as `iw-home\local\config\metadata-rules.cfg.example`.



```
<?xml version="1.0" encoding="UTF-8" ?> ← International Encoding 1  
  
<metadata-rules>  
  <cond vpath-regex="."> ← Vpath Identifier 2  
    <rule name="Default Rule" /> ← Rule Identifier 3  
  </cond>  
</metadata-rules>
```

Sample metadata-rules.cfg File 1 Notes

- 1. International Encoding:** UTF-8 is an encoding of Unicode, a standard for encoding the character sets of international languages. All web assets should specify their encoding as UTF-8. For details about web asset encoding, see Appendix D, “Internationalization”.
- 2. Vpath Identifier:** Names the vpath (in this case all directories) to which the rule(s) named in the following subelement(s) will be applied.
- 3. Rule Identifier:** Names the rule that applies to the preceding vpath. The rule itself is defined in the `<ruleset>` element in `iw-home\local\config\datacapture.cfg`. In this example, the Default Rule rule defined in `datacapture.cfg` will always apply to all directories.

Sample metadata-rules.cfg File 2

The following metadata-rules.cfg file illustrates a more sophisticated example:

```
<metadata-rules>
  <cond vpath-regex="^\default\main\syndication"> ← Vpath Identifier 1
    <rule name="Default" /> ← Rule Identifiers 2
    <rule name="Syndication" /> ←
  <cond vpath-regex="\.pdf$"> ← Vpath Identifier 3
    <rule name="PDF Files" /> ← Rule Identifier 4
  </cond>
  <cond vpath-regex="\.doc$"> ← Vpath Identifier 5
    <rule name="MS Word Files" /> ← Rule Identifier 6
  </cond>
</cond>

<cond vpath-regex="^\default\main\www"> ← Vpath Identifier 7
  <rule name="Default" /> ← Rule Identifiers 8
  <rule name="Web Content" /> ←
  <cond vpath-regex="\.html$"> ← Vpath Identifier 9
    <rule name="HTML Files" /> ← Rule Identifier 10
    <cond vpath-regex="\\pr\\"> ← Vpath Identifier 11
      <rule name="PR" /> ← Rule Identifier 12
    </cond>
    <cond vpath-regex="\\corp\\"> ← Vpath Identifier 13
      <rule name="Corporate" /> ← Rule Identifier 14
    </cond>
  </cond>
</cond>
</metadata-rules>
```

Sample metadata-rules.cfg File 2 Notes

- 1. Vpath Identifier:** Files on the \main\syndication branch will always receive the rules named in the following subelements.
- 2. Rule Identifiers:** The Default and Syndication rules defined in datacapture.cfg will always apply to the \main\syndication branch.

3. **Vpath Identifier:** Files ending in `.pdf` on the `\main\syndication` branch will receive rules in addition to those defined by Default and Syndication.
4. **Rule Identifier:** The PDF Files rule defined in `datacapture.cfg` will apply to files ending in `.pdf` on the `\main\syndication` branch.
5. **Vpath Identifier:** Files ending in `.doc` on the `\main\syndication` branch will receive rules in addition to those defined by Default and Syndication.
6. **Rule Identifier:** The MS Word Files rule defined in `datacapture.cfg` will apply to files ending in `.doc` on the `\main\syndication` branch.
7. **Vpath Identifier:** The `\main\www` branch will always receive the rules named in the following subelements.
8. **Rule Identifiers:** The Default and Web Content rules defined in `datacapture.cfg` will apply to the `\main\www` branch.
9. **Vpath Identifier:** Files ending in `.html` on the `\main\www` branch will receive rules in addition to those defined by Default and Web Content.
10. **Rule Identifier:** The HTML Files rule defined in `datacapture.cfg` will apply to files ending in `.html` on the `\main\www` branch.
11. **Vpath Identifier:** Files ending in `.html` in the `pr` directory on the `\main\www` branch will receive rules in addition to those defined by Default and Web Content.
12. **Rule Identifier:** The PR rule defined in `datacapture.cfg` will apply to files ending in `.html` in the `pr` directory on the `\main\www` branch.
13. **Vpath Identifier:** Files ending in `.html` in the `corp` directory on the `\main\www` branch will receive rules in addition to those defined by Default and Web Content.
14. **Rule Identifier:** The Corporate rule defined in `datacapture.cfg` will apply to files ending in `.html` in the `corp` directory on the `\main\www` branch.

Configuring datacapture.cfg

The `datacapture.cfg` file defines rule sets for capturing data. Rules are referred to by name in `metadata-rules.cfg` (see “Configuring metadata-rules.cfg” on page 191).

Rules contain *items*, where each item is a single set of data that is to be captured from the end-user. An item consists of one or more *instances*. Each instance encapsulates how to capture the data for the item, and each instance defines an ACL that determines which (if any) instance a particular user is allowed to use to enter the data.

The metadata capture form is a data capture template (DCT) that is configured specifically for metadata capture. The DCT subsystem that generates the metadata capture form is the same subsystem that generates DCTs for TeamSite Templating. A major difference between the two implementations is the location of the `datacapture.cfg` file. TeamSite Templating relies on multiple `datacapture.cfg` files (one for each data type), while metadata capture relies on a single `datacapture.cfg` file (in `iw-home\local\config`).

See “Setting Up Data Capture Templates” in the *TeamSite Templating Developer’s Guide* for a complete explanation of `datacapture.cfg` files, including annotated examples and explanations of elements and attributes. Note that even though the examples in the *TeamSite Templating Developer’s Guide* are specific to TeamSite Templating, they are useful as reference points for setting up `datacapture.cfg` for metadata capture.

An example `datacapture.cfg` file configured specifically for metadata capture is also included with TeamSite (refer to `iw-home\local\config\datacapture.cfg.example`). An annotated explanation of that file is shown in “Sample datacapture.cfg File 1” on page 198.

When you set up `iw-home\local\config\datacapture.cfg` for your site, it is recommended that you copy and edit the `datacapture.cfg.example`, using the following DTD and annotated examples as reference points for your own site-specific configuration.

DTD: datacapture.cfg

The `datacapture.cfg` file uses the following DTD. This DTD is also available online in `iw-home\local\config`.

```
<!ELEMENT data-capture-requirements (ruleset)*>
<!ATTLIST data-capture-requirements
  name CDATA #REQUIRED
  type(metadata|content|workflow) #REQUIRED
>

<!ELEMENT ruleset (item)*>
<!ATTLIST ruleset
  name CDATA #REQUIRED
>

<!ELEMENT item (database?,(%instance;)*)>
<!ATTLIST item
  name CDATA #REQUIRED
  description CDATA #IMPLIED
>

<!ENTITY % instance "(checkbox|radio|text|textarea|select|replicant)" >

<!ELEMENT checkbox(allowed|option)*>
<!ATTLIST checkbox
  required (t|f)"f"
  delimiter CDATA", "
>
<!ELEMENT radio (allowed|option)*>
<!ATTLIST radio
  required (t|f) "f"
>
<!ELEMENT text (allowed)*>
<!ATTLIST text
  required (t|f) "f"
  maxlength NUM
  size NUM
  validation-regex CDATA -- regex(5) for validating this element -
>
```

```

<!ELEMENT textarea (allowed)*>
  <!ATTLIST textarea
    required (t|f) "f"
    rows NUM
    cols NUM
    validation-regex CDATA -- regex(5) for validating this element -
  >
<!ELEMENT select (allowed|optgroup|option)*>
  <!ATTLIST select
    required (t|f) "f"
    size NUM
    multiple (t|f) "f"
    delimiter CDATA ", "-- for multiple=t only --
  >
<!ELEMENT replicant (allowed|item)*>
  <!ATTLIST replicant
    min NUM
    max NUM
    default NUM
  >

<!ELEMENT optgroup (optgroup*, option*)+>
  <!ATTLIST optgroup
    label CDATA #REQUIRED
  >

<!ELEMENT option EMPTY>
  <!ATTLIST option
    selected (t|f) "f"
    value CDATA #IMPLIED
    label CDATA #REQUIRED
  >

<!ELEMENT database EMPTY >
  <!ATTLIST database
    deploy-column(t|f) "t"
    searchable (t|f) "t"
    data-type CDATA "VARCHAR(255)"
    data-format CDATA #IMPLIED
  >

```



```
<!ELEMENT allowed (cred|and|or|not)>
```

```
<!ELEMENT cred EMPTY>  
<!--ATTLIST cred  
  role CDATA #IMPLIED  
  user CDATA #IMPLIED  
>
```

```
<!ELEMENT and (cred|and|or|not)+>
```

```
<!ELEMENT or (cred|and|or|not)+>
```

```
<!ELEMENT not (cred|and|or|not)>
```

Sample datacapture.cfg File 1

The following `datacapture.cfg` file is distributed with TeamSite as `iw-home\local\config\datacapture.cfg.example`. See the section immediately following the file for an explanation of the numbered callouts. See the *TeamSite Templating Developer's Guide* for a complete explanation of `datacapture.cfg` files.

```
<?xml version="1.0" encoding="UTF-8" ?> ← International Encoding 1
```

```
<!-- A <data-capture-requirements> element with type="metadata"
      can contain multiple <ruleset> elements.
```

Note: The <database> elements have no effect on the metadata capture process. These optional elements are used to help integrate metadata capture with Data Deploy. Data Deploy configuration files can be automatically generated from datacapture.cfg files. The <database> tags ensure the database tables are built using the appropriate datatype.

```
-->
```

```
<data-capture-requirements type="metadata"> ← Metadata Identifier 2
```

```
<ruleset name="Default Rule"> ← Rule Identifier 3
```

```
<description>
```

```
    This rule applies to all files on all branches.
```

```
</description>
```

```
<item name="Title">
```

```
    <database searchable="t" data-type="VARCHAR(60)" /> ← database Element 4
```

```
    <text required="t" maxlength="60" /> ← Instance (text) 5
```

```
</item>
```

```
<item name="Description">
```

```
    <database data-type="VARCHAR(100)" />
```

```
    <text required="t" maxlength="100" />
```

```
</item>
```

```
<item name="Type">
```

```
    <database data-type="VARCHAR(30)" />
```

```
    <select>
```

```
        <option label="White Paper" value="white_paper" />
```

```
        <option label="Datasheet" value="datasheet" />
```

```
        <option label="Press Release" value="press_release" />
```

```
        <option label="Architecture Overview" value="architecture" />
```

```
        <option label="Futures Overview" value="futures" />
```

```
        <option label="Program Material" value="program_material" />
```

```
    </select>
```

```
</item>
```



```
<item name="Category">
  <database data-type="VARCHAR(40)" />
  <!-- To use the example callout,
    1. Comment out this select element.
    2. Uncomment the text element.
  -->
  <select>
    <option label="Internet - Financial" value="financial_internet"/>
    <option label="Internet - Manufacturing" value="manufacturing_in
      ternet" />
    <option label="Internet - Services" value="services_internet" />
    <option label="Extranet - Tier 1" value="tier_1_extranet" />
    <option label="Extranet - Tier 2" value="tier_2_extranet" />
    <option label="Extranet - Tier 3" value="tier_3_extranet" />
  </select>
  <!--
  <text>
    <callout type="cgi"
      label="Query for Categories"
      url="/iw-bin/iw_cgi_wrapper.cgi/example_datacapture_callout.i
        pl/metadata-category-options.txt" />
  </text>
  -->
</item>

<item name="Languages">
  <database data-type="VARCHAR(20)" />
  <checkbox>
    <option label="English" value="English" />
    <option label="German" value="German" />
    <option label="French" value="French" />
    <option label="Japanese" value="Japanese" />
    <option label="Chinese" value="Chinese" />
  </checkbox>
</item>

<item name="Source">
  <database data-type="VARCHAR(50)" />
  <text maxlength="50" />
</item>
```



```

<item name="Launch Date">
  <database data-type="DATE" data-format="yyyy-MM-dd" />
  <text required="t" maxlength="10" validation-regex="^[0-9][0-9]
    [0-9][0-9]-[0-1][0-9]-[0-3][0-9]$" />
</item>

<item name="Expiration Date">
  <database data-type="DATE" data-format="yyyy-MM-dd" />
  <text maxlength="10" validation-regex="^[0-9][0-9][0-9][0-9]-[0-1][
    0-9]-[0-3][0-9]$" />
</item>

<item name="Keywords">
  <database data-type="VARCHAR(100)" />
  <text maxlength="100" />
</item>
</ruleset>
</data-capture-requirements>

```

DATE datatype ⁶

validation-regex ⁷

Sample datacapture.cfg File 1 Notes

The following information is specific to the example file shown in the preceding section. For more detailed information about `datacapture.cfg` files, see the *TeamSite Templating Developer's Guide*.

- 1. International Encoding:** UTF-8 is an encoding of Unicode, a standard for encoding the character sets of international languages. All web assets should specify their encoding as UTF88. For details about web asset encoding, see Appendix D, "Internationalization".
- 2. Metadata Identifier:** When configuring `datacapture.cfg` for metadata capture, you must specify "type=metadata" in the `<data-capture-requirements>` element as shown here.
- 3. Rule Identifier:** The `<ruleset>` element contains all of the items that make up the rule set that defines the appearance and behavior of the data capture form. A `datacapture.cfg` file that is configured for metadata capture can contain any number of `<ruleset>` elements (as opposed to TeamSite Templating `datacapture.cfg` files, which can contain just one `<ruleset>` element). This example file happens to contain just one `<ruleset>` element; it could contain more if necessary. The rule defined here is named `Default Rule`, and is referenced by the `metadata-rules.cfg` file shown page 192. The name attribute is required and its value appears in the TeamSite GUI as the name of the data capture form. More than one form can appear on a single page. Optional subelements `<label>`, `<description>`, `<item>`, and `<itemref>`. The `<label>` subelement is used to provide a label on the data capture form. The `<itemref>` subelement requires the name attribute and is used as a stand-in for the `<item>` subelement in a `<symbol-table>` element.
- 4. database Element:** The optional `<database>` element facilitates the use of the appropriate data type in DataDeploy and is used only for generation of the `mdc_dd.cfg` file. It does not control any aspects of the metadata capture or search forms. The `<data-type>` and `<searchable>` information in the `<database>` element are passed on to `mdc_dd.cfg`, which in turn uses that information to control how metadata is deployed to a database via DAS. The `<database>` element has four attributes. Because attribute order in XML documents is important; these attributes should occur in the order they are listed:

- `deploy-column` can be either "t" (default) or "f" and allows you to set whether or not data entered for the item is deployed to a database column.
- `searchable` can be either "t" (default) or "f" and allows you to set whether or not users can search against this item.
- `data-type` is required and is any JDBC database type. If you do not set the `data-type` attribute, a default datatype of VARCHAR (255) is set in `mdc_dd.cfg`.
- `data-format` describes the format if date or time is specified for the `data-type` attribute (see callout 6). If a value for `data-format` is specified, the instance should contain a validation regex to force a valid entry in the field (see callout 7).

In this example, `deploy-column` would be the first attribute if it were set. It is not, so all input for this item will be deployed to a database column. Next, the `searchable` attribute is specified as "t"; however, because this is the default value for the attribute, `searchable` need not be included here. Following `searchable` is `data-type`, here specifying the input to be stored as a string. If date or time is set as the value for the `data-type`, a `data-format` attribute should end the element.

5. Instance (text): The optional `<text>` element controls the length of text entry fields in metadata capture and search forms. It also controls whether an end user is required to enter text in a field. If the datatype is `date` or `time` and a format has been specified, it is best to include a `validation-regex` to force users to input data in the correct format (see callout 7). In this example the user is required enter a string of between one and 60 characters in the text field. The data entered for this item is stored in the database as VARCHAR and is searchable.

6. DATE datatype: If the datatype is set to `date` or `time`, it is recommended you specify a `data-format` and include a validation. Because there are many formats for date and time, specifying a format forces the user to enter data in that format and reduces the chance of user error. The value for `data-format` can be any valid Java format for a date or time.

7. validation-regex: The user can be forced to enter a date or time in the format you specify by including a validation regex. The value for the `validation-regex` attribute must match the format specified in for `data-format`. The regex in this example specifies the range of digits that can be entered for `yyyy-MM-dd` and that dashes must separate year, month and day. The following table shows validation regex examples for several



supported datatypes. The `<database>` and `<text>` elements shown in the table are subelements of the `<item>` element. Some regex lines are wrapped due to formatting constraints. You should enter them all on one line in your configuration file.

Datatype	Notes	Example
DATE	If data-type is DATE, the data-format must be a format string that is valid for the Java simple date format class. Formats do not have to be year-month-day, any valid format will work.	<pre><database data-type="DATE" data-format="yyyy-MM-dd" /> <text maxlength="10" validation-regex="^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]\$" /></pre>
INT	Allows any integer up to 7 digits. This example assumes that you want to store data as integers, not dollars and cents.	<pre><database data-type="INT" /> <text maxlength="7" validation-regex="^[0-9]\{0,\}\$" /></pre>
REAL	Allows any decimal up to 8 digits (including decimal). The regex allows 0 or more digits, followed by a decimal point, followed by zero or more digits.	<pre><database data-type="REAL" /> <text required="t" maxlength="8" validation-regex="^[0-9]\{0,\}\.[0-9]\{0,\}\$" /></pre>

Sample datacapture.cfg File 2

The following datacapture.cfg file is written to work with the file shown earlier in “Sample metadata-rules.cfg File 2” on page 193.

```
<!-- This config file defines rulesets for capturing data.
Rules are referred to by name in other config files, such as
metadata-rules.cfg. Rules contain "items"; one item is a single
(set of) data that is to be captured from the end user.
An item consists of one or more "instances". Each instance
encapsulates how to capture the data for the item, and each
instance defines an ACL that determines which (if any)
instance a particular user is allowed to use to enter the
data. Instances are text, textarea, radio, checkbox, select,
and replicant. (Others are coming.)
Replicants are very special kinds of instances; they are
repeatable. Replicants contain _items_ instead of just an ACL
like the other types of instances.
-->
<data-capture-requirements type="metadata">
  <ruleset name="Default">
    <item name="Author">
      <database data-type="VARCHAR(12)" />
      <!-- This item is represented by a text box. -->
      <text size="12" required="t" />
      <!-- no ACL means open access for everyone -->
    </item>
  </ruleset>
  <ruleset name="Syndication">
    <item name="Category">
      <database data-type="VARCHAR(10)" />
      <!-- This item is represented by a series of four
checkboxes. -->

      <checkbox required="t" delimiter="/"> ← Instance (checkbox) with delimiter 1

      <!-- We want the TeamSite extended attribute
to use "/" as the value delimiter when
concatenating all the selected values,
e.g., "Partners/Customers." -->

      <option label="Partners" />
      <option label="Suppliers" />
      <option label="Customers" />
      <option label="Internal" />
```



```
<ruleset name="PDF Files">
  <item name="All Keywords">
    <!-- All nested <item> elements must be "type compatible"
         if the fields are going to be deployed to a database. -->
    <database data-type="VARCHAR(20)" searchable="f" />
    <!-- Because any number of keywords may apply to a
         single file, we use a replicant instance for
         the "keywords" item. -->
    <replicant default="3" min="1" max="12"> ← Instance (replicant) 2
      <!-- We allow from 1 to 12 keywords. -->
      <!-- This replicant instance contains just one item,
           which has two instances. -->
      <!-- When there are multiple instances, the first
           instance whose ACL allows the current user
           will be the instance used for that user. -->
      <item name="Keyword">
        <text size="20" required="t">
          <!-- This ACL allows "joe" and masters
               to type anything she wants. -->
          <allowed> ← Access Control Limiter (ACL) 3
            <or>
              <cred user="joe" /> ← Access Identifier 4
              <cred role="master" />
            </or>
          </allowed>
        </text>
        <select required="t">
          <!-- Everyone but joe has to choose from
               pre-determined choices. -->
          <allowed>
            <not>
              <cred user="joe" />
            </not>
          </allowed>
          <option label="supply chain" />
          <option label="marketing" />
          <option label="sales promotions" />
          <option label="earnings" />
          <option label="facilities" />
          <option label="eCommerce" />
        </select>
      </item>
    </replicant>
  </item>
</ruleset>
```

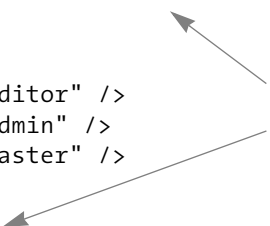
```

<ruleset name="MS Word Files">
  <!-- ... -->
</ruleset>
<ruleset name="Web Content">
  <!-- ... -->
</ruleset>
<ruleset name="HTML Files">
  <!-- ... -->
</ruleset>
<ruleset name="PR">
  <item name="Go-Live Date">
    <!-- The database column must allow strings like "Today" and
         "Tomorrow", so the DATE datatype cannot be used.
         This configuration decreases the usefulness of searching
         this database column -->

    <database searchable="f" data-type="VARCHAR(10)" />

    <!-- This text instance has a regular expression that
         determines validity of user-entered data.
         In this case, the regex requires the user
         to enter "##/##/####". -->
    <text size="10" validation-regex="^[0-9][0-9]/[0-9][0-9]/[0-9][
    0-9] [0-9][0-9]$" >
      <allowed>
        <or>
          <cred role="editor" />
          <cred role="admin" />
          <cred role="master" />
        </or>
      </allowed>
    </text>
    <select required="t">
      <option label="Today" />
      <option label="Tomorrow" selected="t" />
      <option label="Next Week" />
    </select>
    <allowed>
      <cred role="author" />
    </allowed>
  </item>
</ruleset>
<ruleset name="Corporate">

```



Variable Instances ⁵



Sample datacapture.cfg File 2 Notes

The following information is specific to the example file shown in the preceding section. For more detailed information about `datacapture.cfg` files, see the *TeamSite Templating Developer's Guide*.

- 1. Instance (checkbox or select) with delimiter:** Specifies the delimiting character used when data from all check boxes is concatenated by the data capture subsystem. The default delimiter is a comma (,). In this example a “/” is used as the delimiter to separate the concatenated values for the checked `<option>` elements.
- 2. Instance (replicant):** Specifies a repeatable instance that can contain multiple nested items and instances. When there are multiple instances, the first instance whose ACL allows the current user to enter data will be the instance used for that user. `<replicant>` is the only instance that can contain nested items and instances. Whenever additional iterations of the instance can be displayed (that is, if the `max` threshold has not yet been reached), an **File > Add Above** and **File > Add Below** menu items are active. Whenever iterations of the instance can be removed (that is, if the `min` threshold has not yet been reached), The **File > Delete** menu item is active. If a `<replicant>` has four items, the **Add** menu item displays another set of four items in the data capture form. In this example, if the user's username is `joe` or role is `master`, three keyword text fields display; if the user is not a master or “joe”, then three drop-down selection boxes display. Keyword instances can be added or removed to a minimum of one and a maximum of 12 using the **Add** or **Delete** options in the **File** menu.
- 3. Access Control List (ACL):** The `<allowed>` element lets you set an ACL to specify which users can or cannot use a specific instance to enter data. If `<allowed>` is not set, the instance is visible to and can be used to input data by any user. The allowed element can have any of the following elements:
 - `<cred>` lets you name a user or role in the ACL.
 - `<and>` defines multiple users or multiple roles that can use the instance.
 - `<or>` defines users and roles that can use the instance.
 - `<not>` defines a user or role that is not allowed to use the instance. The instance does not display for users not allowed to use that instance.

In this example **Keyword** text fields display for the user with username `joe` or the role `master`; for other users only **Keyword** selection drop-down menus display.

4. **Accessor Identifier:** The `<cred>` element is a child element of `<and>`, `<not>`, or `<nor>` and lets you identify the accessor by role and username. Note that `<cred>` requires exactly one attribute, either `role` or `user`. In this example two `<cred>` elements are combined under the parent `<and>` element so that the ACL applies to both the username `joe` and role `master`. Text fields display for users with either identification, while drop-down selection menus display for others.
5. **Variable Instances:** Within an `<item>` an instance can be made available to certain users or roles and not to others. In this example all Editors, Administrators, or Masters are offered text fields and can input variable text strings, while Authors are offered drop-down menus with predefined choices. Note that because there is one `<database>` element for each `<item>`, and that input for this item could be either dates or a character string, the datatype must be set to `VARCHAR`. It is not recommended that you create a `VARCHAR` database in which numerical data, such as dates or time, might be stored; operands useful for retrieving dates and time such as “between”, “less than”, “greater than” cannot be used to search such information.

Adding Metadata Capture to the TeamSite GUI

Because metadata capture is a file-specific feature, it is recommended that end users access it via the **File** menu in the TeamSite GUI. To add a **Set Metadata** item to the TeamSite GUI's **File** menu, add the following line to the `[iwcgi]` section of `iw.cfg`:

```
custom_menu_item_metadata="File", "Set Metadata", "iwmetadata.cgi"
"all" "width=800,height=570,scrollbars=yes,resizable=yes"
```

This line specifies the following:

- The TeamSite GUI menu (**File**) to which the item will be added.
- The name of the new item (**Set Metadata**).
- The CGI (`iwmetadata.cgi`) that will execute when the item is selected.
- Which users (`all`) can see the menu item.
- The appearance and behavior of the window in which the CGI runs.

See “Custom Menu Items” on page 125 for more information about adding and enabling custom menu items.

Metadata Capture End Result

After you configure metadata capture, end users can access it via the TeamSite GUI to set metadata on files. The end result of a metadata capture session is the addition of TeamSite extended attributes to one or more files. For example:

File: \default\main\www\WORKAREA\jk\pr\BigAnnouncement.html

Name	Value
TeamSite\Metadata\Author	jk
TeamSite\Metadata\Go-Live Date	07/04/2000

File: \default\main\syndication\WORKAREA\bill\fall2000.pdf

Name	Value
TeamSite\Metadata\Author	bill
TeamSite\Metadata\Category	Partners\Customers\Internal
TeamSite\Metadata\Category\Partners	Y
TeamSite\Metadata\Category\Customers	Y
TeamSite\Metadata\Category\Internal	Y
TeamSite\Metadata\Keywords\0\Keyword	supply chain
TeamSite\Metadata\Keywords\1\Keyword	earnings
TeamSite\Metadata\Keywords\2\Keyword	eCommerce

These are the extended attributes that would be displayed via the TeamSite GUI from the **File > File Properties** menu. These extended attributes can now be deployed to a database via DataDeploy. the deployment can be manual, or automatic through Database Auto-Synchronization (DAS). See the *DataDeploy Administration Guide* for more information.

Metadata Capture and TeamSite Workflow

The following sections describe key interactions between metadata capture and TeamSite workflow.

Specifying Files in Workflow Tasks

Metadata capture includes the ability to self-filter a list of files on which to capture metadata. For example, a user task or a job task can name a set of files upon which metadata will be set. All symlinks, directories, and deleted files that are part of the file set will be filtered out and ignored. Only actual files will have metadata set.

Initiating Metadata Capture from a Job Specification File

The following sample `<cgitask>` section from a job specification file shows the syntax necessary to initiate a typical metadata capture process from within a job. The task owner in this example is `jk`. The task in this example is associated with the area shown in `areavpath`. See the *TeamSite Workflow Developer's Guide* for the `<cgitask>` DTD and other general information.

```
<cgitask name="metadata" owner="jk">
  <description>apply metadata.</description>
  <areavpath v="\default\main\test\WORKAREA\jk" immediate ="+"/>
    <successors>
      <successorset description="set">
        <succ v="confirm" />
      </successorset>
    </successors>
    <command v="\iwmetadata.cgi" />
    <activation>
      <or>
        <pred v="start" />
      </or>
    </activation>
</cgitask>
```

Metadata Search

The following sections describe:

- A metadata search overview.
- The prerequisites for configuring metadata search.
- The main components that make up metadata search.
- How to configure metadata search.

Overview

The metadata search subsystem uses search parameters supplied by an end user via a search form to query a database containing metadata. The end result is a list of files, displayed in the TeamSite GUI, that contain metadata tags matching the search parameters. The search form is based on configuration files also used by metadata capture and generated by DAS. This relationship ensures that the search form contains fields only for data that is already stored in the metadata database. Details about these files are presented later in this section.

Metadata search is an area-specific feature. That is, it performs a search for metadata tags on files in the entire area and all subareas from which it was executed. For example, if you execute metadata search from `\default\main\www\WORKAREA\w1`, all files in `w1` and its subdirectories are searched. If you execute metadata search from `\default\main\www\WORKAREA\w1\marketing`, all files in `marketing` and its subdirectories are searched. You can execute metadata search from a workarea or any subdirectory within a workarea. You cannot execute it from a staging area, edition, or branch. See the *TeamSite User's Guide* for more information about metadata search usage.

Prerequisites

It is essential that you configure the following features before configuring and running metadata search:

- Metadata capture, as described earlier in this chapter. This is required because metadata search relies on the same `datacapture.cfg` file as metadata capture. If this file is not configured correctly, metadata search will not run.

- DataDeploy’s Database Auto-Synchronization (DAS) module as described in the *DataDeploy Administration Guide*. This is required because metadata search relies on the `mdc_dd.cfg` file, which is generated automatically when DAS is configured. If DAS is not configured correctly, metadata search will not run.

In addition, you must already have deployed metadata to a database via DAS before running metadata search. This is required because metadata search searches the database specified in the DAS configuration files. It does not search the TeamSite backing store or any database not specified in the DAS configuration files.

If all of these prerequisites are met, you can proceed with the metadata search configuration as described in “Configuring Metadata Search” on page 215. It is recommended that you read the following “Components” section before performing the configuration.

Components

Metadata search relies on six main components:

- The same `iw-home\local\config\datacapture.cfg` configuration file used by the metadata capture subsystem.
- The DataDeploy configuration file `iw-home\local\config\mdc_dd.cfg`, which is generated automatically when you configure DataDeploy Database Auto-Synchronization (DAS) or when you execute the `iwsyncdb.ipl -mdcddgen` command.
- The metadata search CGI `iwsearchmetadata.cgi`, which interprets data from end users and rules in `datacapture.cfg` and `mdc_dd.cfg`, and produces browser graphics and prompts.
- The `[valid_search_paths]` section of `iw-home\etc\iw.cfg`.
- A browser interface for end-user input.
- The database containing metadata deployed via DAS.

The following diagram shows how these components work together. Sections following the diagram explain each diagram step and component in detail.

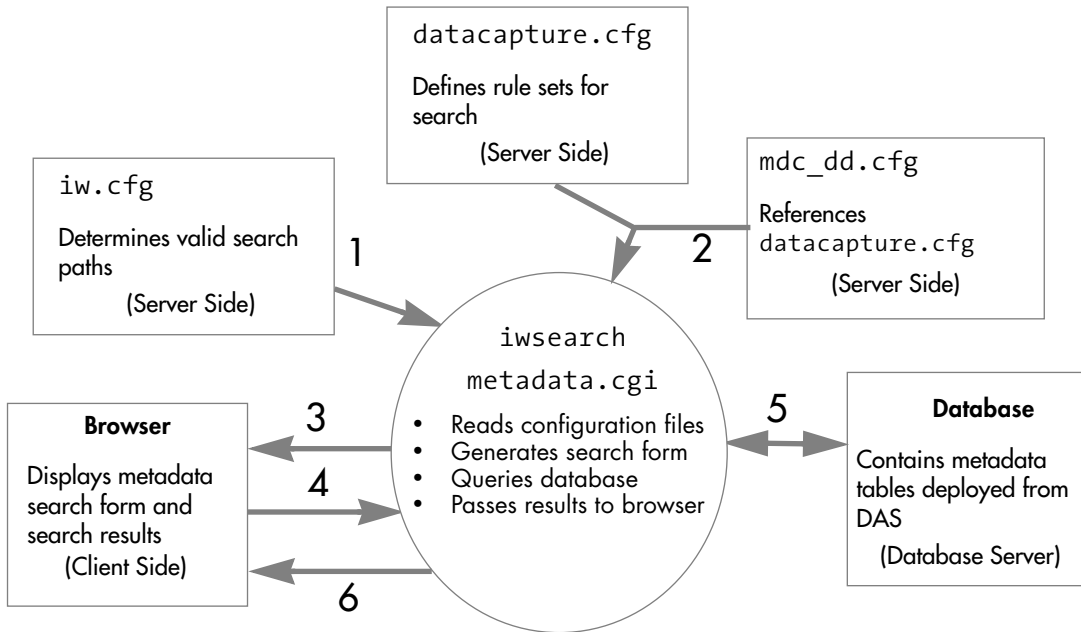


Diagram Key

1. The metadata search CGI reads the [valid_search_paths] section of the *iw-home\etc\iw.cfg* configuration file to determine the paths where metadata search is valid. By default, all paths are considered valid search paths. For more information see “Changing Valid Search Paths” on page 217.
2. The metadata search CGI reads the *mdc_dd.cfg* and *datacapture.cfg* configuration files. The information from these files is used by the search CGI to determine what should be displayed in the metadata search form. The information provided by *mdc_dd.cfg* controls whether a metadata field is searchable, what label each field has, and which operators are valid for each field in the search form. For example, if a metadata tag residing on the database uses a data type of CHAR, the search form will contain operators such as Contains, Does contain, etc. for that specific field. The end user can then select one of these field-specific operators to set the search parameters for that field. However, if a metadata tag uses a data type of INTEGER, the search form will contain operators including Equals, Does not equal,

for that specific field. By using both `datacapture.cfg` and `mdc_dd.cfg`, the search CGI ensures that each field in a search form will always contain the appropriate set of operators from which an end user can choose.

See “Configuring Metadata Search” on page 215 for more information.

3. The metadata search CGI displays the search form on the client system via the GUI.
4. An end user fills in search parameters and submits the search form back to the metadata CGI.
5. The metadata search CGI constructs the appropriate query statements and queries the database.
6. The search results are displayed in the TeamSite GUI.

Configuring Metadata Search

You must perform two main activities to configure metadata search:

1. Ensure that metadata capture and DAS are synchronized.
2. Add a **Search Metadata** item to the TeamSite GUI so that end users can access metadata search.

The following sections describe these steps in detail. Additional configuration information is included in subsequent sections in case you need to customize the metadata search form or other characteristics of metadata search.

Synchronizing Metadata Capture and DAS

Metadata search relies on correct synchronization of metadata capture and DAS. If these two features are not synchronized, metadata search will not run correctly. The issue is as follows:

When you configure DAS, you execute the `iwsyncdb.ipl -initial` command. This command generates several files, including `mdc_dd.cfg`. The `mdc_dd.cfg` file is in turn based on information from `datacapture.cfg`. The `datacapture.cfg` file must be configured specifically for metadata capture at your site to ensure that `mdc_dd.cfg` is generated correctly for use with metadata capture and search at your site. Therefore, it is *essential* that `mdc_dd.cfg`

be generated *after* you set up `datacapture.cfg` in `iw-home\local\config` as described earlier in this chapter. There are two ways to ensure that this is the case:

1. Configure metadata capture as described earlier in this chapter before configuring DAS as described in the *DataDeploy Administration Guide*, or
2. Execute the following command if DAS was already configured prior to your configuring `iw-home\local\config\datacapture.cfg`:

```
iwsyncdb -mdcddgen [-force]
```

You can execute this command whenever you need to resynchronize DAS and metadata capture/search. See the *DataDeploy Administration Guide* for details about `iwsyncdb` usage.

Note: Regenerating `mdc_dd.cfg` overwrites the existing version of the file, including any changes you might have made to it.

Adding Metadata Search to the TeamSite GUI

Because metadata search is an area-specific feature, it is recommended that end users access it via the **View** menu in the TeamSite GUI. To add a **Search Metadata** item to the TeamSite GUI's **View** menu, add the following line to the `[iwcgi]` section of `iw-home\etc\iw.cfg`:

```
custom_menu_item_searchea="View", "Search Metadata", "iwsearchmetadata.cgi",  
"all", "scrollbars=yes,resizable=yes,width=640,height=545"
```

Note: Due to space limitations, this line appears to wrap. The line in the configuration file must not wrap.

The preceding line specifies the following:

- The TeamSite GUI menu (**View**) to which the item will be added.
- The name of the new item (**Search Metadata**).
- The CGI (`iwsearchmetadata.cgi`) that will execute when the item is selected.
- Which users (`all`) can see the menu item.
- The appearance and behavior of the window in which the CGI runs.

See “Custom Menu Items” on page 125 for more information about adding and enabling custom menu items.

Changing Valid Search Paths

Valid paths for metadata search are set in the `[valid_search_paths]` section of `iw.cfg`. By default, all paths are searchable. You can use regular expressions to specify that only certain paths are searchable. See comments in `iw.cfg` for more information.

Making Individual Fields Non-Searchable

You can specify whether any field in a DCT is searchable. By default, all fields are searchable. To make a field non-searchable, specify `searchable="f"` the `<database>` element for that field in `datacapture.cfg`. If you make this change after `mdc_dd.cfg` was generated, you must regenerate it via the `iwsyncdb -mdcddgen` command.

Note: It is not advisable to edit `mdc_dd.cfg` itself. Such action could result in inconsistencies between DAS and metadata capture/search.

Chapter 7

Managing the TeamSite Server

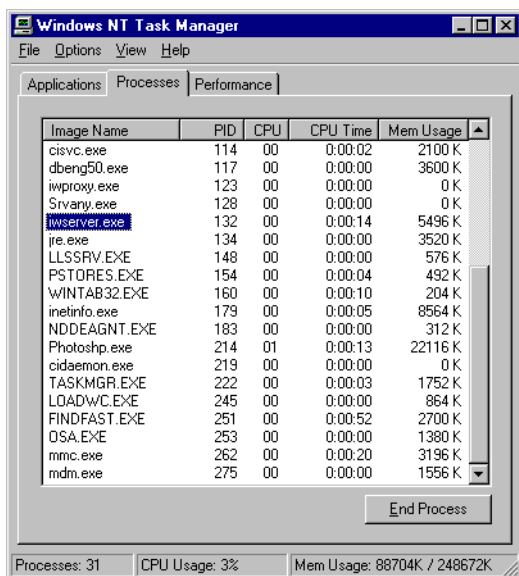
The following topics are described in this chapter:

- [Checking Server Status \(page 220\)](#)
- [Reviewing TeamSite Logs \(page 222\)](#)
- [Monitoring the Server Load \(page 223\)](#)
- [Starting and Stopping the Server \(page 223\)](#)
- [Managing the OpenAPI Server \(page 224\)](#)
- [Reconfiguring iwwebd to Recognize a New IP Address \(page 225\)](#)
- [Re-Encrypting User Authentication Information \(page 226\)](#)
- [Troubleshooting \(page 226\)](#)
- [Managing Server Resources \(page 232\)](#)

Checking Server Status

Verifying Server Operation

Verify that the TeamSite server is running correctly by checking the Windows NT Task Manager:



The Windows NT Task Manager

Make sure that only one `iwserver` is running, and check to see how much memory it is using. To change the amount of memory it is using, edit the `cachesize` line in the `[iwserver]` section of `iw.cfg`. The value of `cachesize` is not the amount of memory that is used, but the number of objects kept in the cache. For more information about the `cachesize` line, see “Cache Size” on page 155.

Checking Request Handling

Verify the server is answering requests correctly by issuing the command:

```
>iw-home\bin\iwversion
```

You will see a response similar to this:

```
iwserver: 5.5.1 Build 6038 Interwoven 20010420
```

If the server does not respond or stops, then the server is not handling requests correctly. Restart the server, as described on page 223.

Verifying the Server Mount

Use Windows Explorer to verify that the Y: (default) drive is mounted as an IFS volume. If you have not used the default location, check the `iwmount` line in the `[locations]` section of `iw-home/etc/iw.cfg` to find out what drive letter you have used. Use Windows Explorer to verify that that drive is mounted as an IFS volume. If it is not, reboot the server.

If you are using IIS, you might find that the Microsoft Management Console shows an Error flag next to the IFS volume when you reboot the Windows NT server. This does not necessarily indicate an error. Because IIS starts before TeamSite, it cannot find the IFS volume when it first starts, and it does not remove the error even after TeamSite starts and the IFS volume appears. You will note that once the TeamSite server does come up, you can navigate into the IFS volume and see the contents of your website.

Finding the Installation Directory

To find the TeamSite installation directory, use the Find command:

1. Select **Find > Files or Folders** from the **Start** menu.
2. Search for **Interwoven** or **TeamSite**. If neither of those searches finds the installation directory, search for `iwserver`.

You can also use the `iwgethome.exe` CLT if it has been installed in your path.

Usage:

```
iwgethome [-h|-v|-o]
```

-h	Displays usage message.
-v	Displays version.
-o	Returns original factory setting value.

Example:

```
>iwgethome
```

returns

```
C:\Program Files\Interwoven\TeamSite
```

Reviewing TeamSite Logs

TeamSite records events in the Windows NT Event Viewer and TeamSite log files as described below.

Windows NT Event Viewer

You can configure TeamSite to log and display any of the following events in the Windows NT Event Viewer:

- DiskLow
- Freeze
- ShutDown
- StartUp
- Thaw

Configuration is controlled by the [iwserver] section in the iw.cfg file.

See the Windows NT Event Viewer documentation provided by its manufacturer for usage details.

TeamSite Log Files

TeamSite records various system events and activities in log files.

Log file	Default location	Contents
Trace log	<i>iw-home</i> \local\logs\iwtrace.log	Record of any irregularities on the TeamSite server. Used by Interwoven Professional Services to diagnose system performance issues. You can find this file using the CLT <code>iwgettrace.exe</code> .
Event log	<i>iw-home</i> \local\logs\iwevents.log	Record of activities on TeamSite. Tracks when files are submitted, published, branches created, etc., including DiskLow, Freeze, ShutDown, StartUp and Thaw events. Used with TeamSite triggering scripts. You can find this file using the CLT <code>iwgetelogs.exe</code> .
Workflow log	<i>iw-home</i> \local\logs\iwjoberrors.log	Record of output from workflow runtime diagnostics.

Monitoring the Server Load

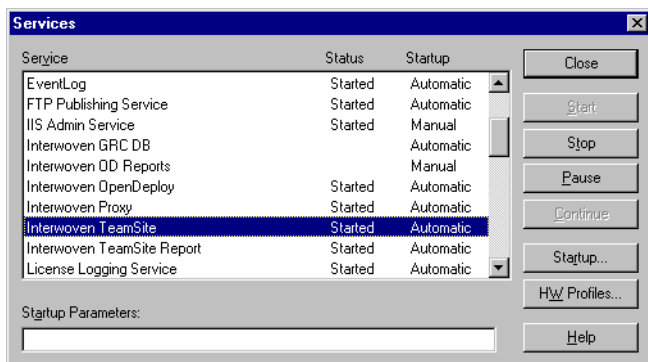
The TeamSite CLT `iwstat.exe` returns a list of all current TeamSite processes. See *TeamSite Command-Line Tools* for details about `iwstat.exe` usage and output.

Starting and Stopping the Server

To stop the TeamSite server:

1. Log in to Windows NT with Administrator permissions.
2. Select **Settings > Control Panel** from the Windows NT **Start** menu.

3. Open the **Services** Control Panel.
4. Select **Interwoven TeamSite** from the list of services. Click **Stop**.



Windows NT Services Control Panel

5. Select **Interwoven Proxy** from the list of services. Click **Stop**.
6. If you are using OpenDeploy, select **Interwoven OpenDeploy** from the list of services. Click **Stop**.
7. To restart TeamSite, you must reboot the server. Do not attempt to restart the **Interwoven TeamSite** service from the Services control panel. By default, TeamSite is configured to start automatically on reboot.

Managing the OpenAPI Server

The OpenAPI server is automatically installed and started as part of installing TeamSite, as the Interwoven Servlet Engine depends upon OpenAPI.

Verifying that the OpenAPI Server is Running

The Windows NT Services Control Panel/Windows 2000 Services utility should display “Interwoven OpenAPI Service”, running as Administrator, with status “started”. If the OpenAPI service is not running, then it needs to be restarted as described below.

Starting and Stopping OpenAPI

Windows understands the dependency between the Interwoven OpenAPI and Servlet Engine services. A user attempting to stop the OpenAPI service will be prompted to also stop the Interwoven Servlet. Similarly, starting the Interwoven Servlet will cause Windows to automatically start the OpenAPI service, if it hasn't already been started.

To restart OpenAPI manually, invoke the CLT:

```
>iwreset -a
```

Issuing `iwreset` with this argument makes `iwserver` both reread its configuration and restart OpenAPI, WebDesk, and the Interwoven Administration GUI. Since both WebDesk and the Interwoven Administration GUI depend on OpenAPI, they also need to be restarted when OpenAPI is restarted.

The only time the OpenAPI service should ever be stopped directly is before reinstalling OpenAPI. After re-installing OpenAPI, issue the CLT `iwreset -a` as described above to restart OpenAPI.

For more information about OpenAPI, consult the Administration chapter of the *OpenAPI Developer's Guide*, available online as part of the OpenAPI SDK.

Reconfiguring iwwebd to Recognize a New IP Address

If you change the IP address of the server, you need to reconfigure `iwwebd` to recognize the new address. To do this:

1. Go to the `iwwebd.bin` folder.
2. Run `iwwebd_conf.ipl`.
3. Restart `iwwebd`.

`iwwebd` will be reconfigured and the `iw.cfg` file will be updated with the new IP address.

Re-Encrypting User Authentication Information

The TeamSite CLT `iwsessionkeygen.exe` generates the key used to encrypt user authentication information. Running this command invalidates all current user sessions and generates a new key. You may want to run `iwsessionkeygen.exe` periodically to protect system security.

To generate a new encryption key for user authentication information, log in as Administrator and run `iwsessionkeygen.exe`. All user sessions will need to log in again to continue working in any TeamSite interface.

Note: Since WebDesk Pro does not allow users to easily resume interrupted sessions, it is recommended that you do not run `iwsessionkeygen.exe` while users are working in WebDesk Pro. If you do run `iwsessionkeygen.exe` while users are working in WebDesk Pro, these users may experience a variety of errors (for example, failure to connect to the TeamSite server, or invalid session strings). If this happens, WebDesk Pro users should log out, then log in again.

Troubleshooting

Interwoven's Professional Services and Technical Support can assist you with any installation and configuration issues you might encounter. You can also consult the Interwoven Support Knowledge Base, available at <http://support.interwoven.com>.

Troubleshooting TeamSite Access

One common cause of TeamSite access errors is incorrect user specification in the roles files. Open the roles files and check to make sure that each user is specified as `DOMAIN\user`, not just `user`.

Repairing the Backing Store

The following section contains information about the backing store repair tools provided with TeamSite. If you are experiencing problems with the TeamSite backing store such as missing TeamSite areas or missing file versions, use `iwfsck` to check the backing store. You can use `iwfsck -y` and `iwfsfix` to repair the backing store depending on the results of your backing store check.

iwfsck.exe

Diagnoses backing store problems and allows repair of some of the problems found.

Usage:

```
iwfsck [-h] [-v] [-x|-xx|-xxx] [-l] [-y] [-b path] [-z]
[-d [[-s] | [-f] [-m] [-p]] [-r]]
[-o file] [-e file] [-t file] [-u file] [vpath]
```

- h Displays usage message.
- v Displays version.
- x Requests extra output and increments verbosity level. Prints additional information about what iwbsck is doing as it operates. Each x increments the verbosity level by 1. The highest level of verbosity is level 3 (-xxx). In the higher levels of verbosity, an extremely large quantity of output may be produced.
- l Prints output as HTML. This option is used by the iwfsckcgi.cgi program.
- y Repairs damaged files while running. In this mode, damaged files are deleted while iwfsck is running. The TeamSite server must be down when specifying this option. If the TeamSite server is running when this option is specified, a warning displays and this option is ignored.
- b *path* Uses *path* as the backing store location. The default is the configured backing store location returned by iwgetstore for the TeamSite server.
- z Checks events in branches.
- d Checks directories and files in addition to the normal checking of branches and areas. All directories and files from the *vpath* are walked. If a *vpath* is not specified on the command line, the walk begins at the / *vpath*.

The following options are only allowed when `-d` is specified:

- `-f` Provides a fast reference check (not allowed with `-p`, `-m`, or `-s`). All references from the root are walked aggressively looking for missing references. If a missing reference is found, that part of the tree is marked *suspect*, and a more expensive walk with `vpaths` is done on that part of the tree to determine the directories and files affected by the problem.
- `-s` Provides a stack walk, which is slower but uses less memory than the default (not allowed with `-f`). This mode uses the least amount of memory, but it is the least efficient for walking the entire tree of files and directories from the root.
- `-m` Checks ModLists for directories. A ModList is a data structure that is a shadow tree to the directory structure within a workarea. This shadow tree allows the modified files within a workarea to be determined quickly without having to traverse every file and directory within a workarea.
- `-p` Checks protopath. A protopath is a data structure that allows file names and history information to be determined without the expense of walking up to the root of an area through directories; however, it can be expensive.
- `-r` Checks parents. Parents and anti-parents are the reference counting mechanism used by the TeamSite server. If zero parents are found for a file, it indicates a problem. It can be expensive.

The following options specify where output goes (note that `stdout` and `stderr` may be redirected in the normal way in a command line shell and that `-o` and `-e` are provided to allow redirection when shell redirection is not available):

<code>-o file</code>	Specifies output file for server startup information.
<code>-e file</code>	Specifies the file to write error messages to.
<code>-t file</code>	Specifies the file to write reports to.
<code>-u file</code>	Specifies the summary file.
<code>vpath</code>	Specifies the starting vpath to walk directories when <code>-d</code> is used.

Examples:

To check areas and branches, issue the command:

```
>iwfsck
```

To check directories and files in addition to branches and areas, issue the command:

```
>iwfsck -d
```

Use the following command to check protopath and parents in addition to branches, areas, directories, and files. This command can be very resource intensive.

```
>iwfsck -d -p -r
```

iwfsckcgi.cgi

This program provides an optional GUI interface to run `iwfsck`. You can access this interface through a browser:

`server_name/iw-bin/iwfslogin.cgi`

You will be prompted for the root or Administrator password. Once you have been authenticated, a screen will provide two choices:

- Perform content recovery
- Perform backing store check

To run `iwfsckcgi.cgi`, select **Perform backing store check**.

iwfsfix.exe

If `iwfsck` finds problems that cannot be repaired or the TeamSite server is running when the backing store is diagnosed, it outputs lines in the format:

```
FIX iwfsfix repair args
```

The repairs and their arguments are shown below. To perform necessary repairs, copy the `FIX` line issued by `iwfsck` and paste it on the command line, with the word `FIX` removed.

There are also repairs for ModLists that must be performed when the TeamSite server is running. On Windows NT, the ModList repair lines are in the format:

```
FIX echo x > junkfile; del /f junkfile
```

junkfile is a uniquely named file that is created and removed from an affected directory.

The repairs that can be performed with `iwfsfix` are:

```
delete_tag branch_id tag_id
```

Removes the reference to a tag (lock) from a branch. This is done when the tag point itself is missing.

`delete_tag_and_point branch_id tag_id`

Deletes the reference to a tag (lock) from a branch and removes the tag point itself. This is usually done when a tag duplicates or conflicts with another tag within a branch.

`delete_direntry directory_id diritems_index filename`

Deletes the directory entry for a damaged or missing file.

`replace_direntry directory_id diritems_index filename new_standin_id`

Repairs a directory entry to point to a correct standin ID.

`delete_area area_id`

Deletes the point for an area.

`delete_area_from_branch branch_id area_id workarea | edition`

Deletes the reference to an area (workarea or edition) from a branch. This cannot be done on a staging area because a branch by definition always contains a staging area.

`null_previous point_id`

Sets to null (-1) the PreviousPoint reference within a point. This is done when the PreviousPoint reference for a point is incorrect.

`clone_diritems directory_id diritems_index new_gen_id new_dot_dot`

Clones a set of directory items within a directory to create a new set. This is done when a set of directory items is shared between areas, but it should not be shared.

Managing Server Resources

Shared Directories and the TeamSite Mount Point

If you have shared directories under the TeamSite mount point (default: Y:), they will become unshared after you restart TeamSite. The only directory that will still be shared is Y:\default. To turn on sharing, you must stop the **Server** service in the **Services** control panel, then start it again. All of your previously shared directories under the TeamSite mount point will be shared again.

Changing TeamSite File Locations

The TeamSite shared drive location can be configured to any drive letter by modifying the [locations] section of *iw-home\etc\iw.cfg* (for example, to change the shared drive to X:, edit the [locations] section to contain the line `iwmount=X:`). If you change the shared drive location, you must update the webserver alias (*iw-mount*) accordingly. Changing shared drive locations will require a server reboot.

Enhancing File System Performance on the TeamSite Server

On the TeamSite server only, browsing the TeamSite shared drive via Windows Explorer can be slow. If you are working directly on the TeamSite server, mount a separate Network Drive in Windows Explorer, select the local host (the TeamSite server) and mount the default TeamSite share (IWServer). If you browse the contents of the mounted drive, you will see a markedly improved performance.

Note: Users accessing the TeamSite file system interface remotely (via a network) will not be affected.

Disk Space

TeamSite Data Store

To delete the backing store, shut down the TeamSite server and move or delete the folder that contains the backing store. When you reboot, TeamSite will create a new (empty) backing store when you restart the server. The new backing store will contain only a main branch, which will

have an empty initial edition, a staging area, and no workareas. You can populate this new backing store with content just as you did your old one.

Checking Disk Space Usage

To check the amount of space used by the TeamSite backing store, use Windows Explorer to check the size of the directory returned by

```
>iw-home\bin\iwgetlocation iwstore
```

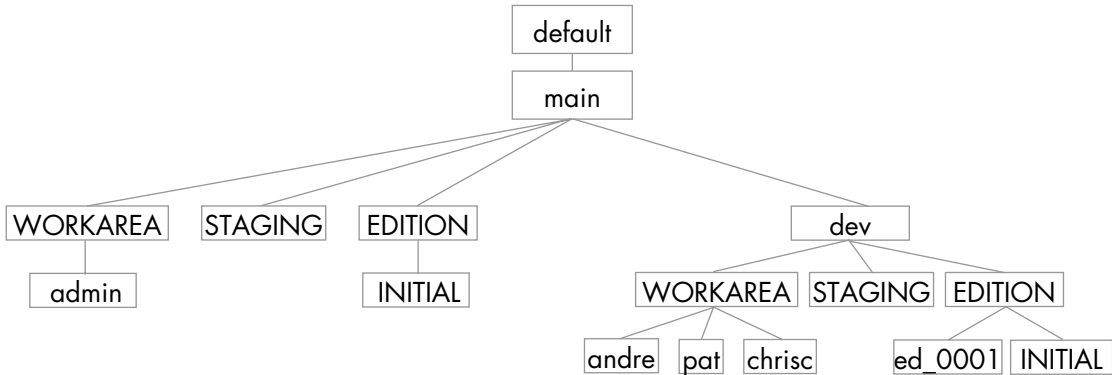
or

```
>iw-home\bin\iwgetlocation iwmount
```

That is, you can check the size of either the backing store *or* the mount point.

TeamSite File System Mount

The TeamSite file system mount contains a file system view of all the branches, workareas, staging areas, and editions on the TeamSite server. TeamSite areas do not contain physical copies of the entire Web site, but rather pointers to the files contained in the Web site. The only physical files contained within TeamSite areas are the files that have actually been modified in those areas. That is, the only files actually contained in a workarea are those files that have been modified in that workarea but not yet submitted; the only files contained in the staging area are the files that have been submitted since it was last published; the only files in an edition are the files that have changed since the previous edition was published.



Sample TeamSite file system structure

Each branch contains three directories: WORKAREA, containing all the workareas on the branch; STAGING, containing the staging area for the branch, and EDITION, containing all editions on the branch. It may also contain directories that hold sub-branches. In the example above, the main branch (main) contains one workarea, a staging area, an initial edition, and a sub-branch (dev). The sub-branch contains three workareas (andre, pat, and chrisc), a staging area, and two editions.

Although many of the files contained within this file system structure are virtual, they can be treated as if they were real. They will appear to exist even when you run links checkers and scripts against them. However, staging areas, editions, and container directories (for example, WORKAREA, EDITION, main, or dev) are all read-only. Only workareas can be written to.

Recovering Disk Space

To reclaim some disk space, you can delete old editions, which will delete all files actually contained in that edition, in addition to all intermediate submissions between publication of editions.

Routine Maintenance: Metadata Forking

Metadata forking conserves disk space by reducing the number of files whose content is duplicated throughout the TeamSite backing store. That is, if you have an old version of a file in one branch, and an identical file version on another branch, the same data may appear twice in the backing store. Metadata forking eliminates this type of duplication. This operation results in no user-visible changes to the TeamSite virtual file system. For example, file histories are unchanged.

To use metadata forking, run the `iwfsshrink` utility. The `iwfsshrink` utility may be run while the TeamSite server is running; however, TeamSite may experience some performance degradation while it is running. Also, `iwfsshrink` may not remove all duplicates (for example, it will not remove any duplicates created by TeamSite users while the utility is running).

1. Issue the `iwfsshrink` command:

```
>iwfsshrink run
```

2. The utility may take several hours to run. Use the `status` option to view the current status. You can also pause the operation with the `pause` option, then restart it with the `run` option. See “`iwfsshrink.exe Syntax`” for a full list of options.

The `iwfsshrink` utility should be run every few months.

iwfsshrink.exe Syntax

<code>iwfsshrink [-h] [-v] [run pause abort status]</code>	
<code>-h</code>	Displays usage message.
<code>-v</code>	Displays version.
<code>run</code>	Starts the <code>iwfsshrink</code> process.
<code>pause</code>	Temporarily stops the <code>iwfsshrink</code> process. It can be restarted with the <code>run</code> option. Because <code>iwfsshrink</code> takes a long time to run, you may want to start it during off-hours. When activity increases, you can pause it until the next period of inactivity.
<code>abort</code>	Terminates the <code>iwfsshrink</code> process.
<code>status</code>	Shows information about the latest <code>iwfsshrink</code> process.

Examples:

```
>iwfsshrink status
```

when `iwfsshrink` has finished running, returns a message similar to:

```
Not currently running.
```

Last started Mon Jun 26 15:47:53 2000
Last completed Tue Jun 27 00:40:04 2000
Files examined: 317974
Bytes examined: 75936814830
Files found to be duplicates: 233430
Files converted: 198352
Bytes removed: 23455046531

Moving the Backing Store and Removing Old Versions

If you are running out of disk space and `iwfsshrink` doesn't recover enough extra space, you might need to move the TeamSite backing store (see page 232). The TeamSite backing store must reside on a single logical volume, e.g., a single disk or an array of disks.

Alternatively, if you have unused branches in TeamSite, you can delete these branches to recover disk space.

Over time, individual branches take up an increasing amount disk space, as the number of versions and files on the branch grows. If you do not need any of your old version history, you can create a new (empty) branch, create a workarea, copy all the old content into the workarea, then delete the old branch. Exercise extreme caution when doing this, as all versioning and metadata information will be irrevocably lost.

Chapter 8

TeamSite Backing Stores

This chapter describes the TeamSite backing store functionality including creating multiple backing stores (either by converting an existing backing store or creating new backing stores). It also includes what you need to know to prepare for conversion. The information is organized as follows:

- Backing Store Overview
- Planning the Backing Store Conversion
- Converting Backing Stores Using the GUI
- Converting Backing Stores from the Command Line
- Creating Multiple Backing Stores
- Administration CLTs
- SID Changes to the TeamSite Backing Store

Backing Store Overview

The backing store is a large directory structure created by the TeamSite installation program that contains TeamSite files and metadata. By default, the backing store is located in `C:\iw-store`.

Previous releases of TeamSite have been limited to one backing store per TeamSite server. This release supports as many as eight backing stores per TeamSite server. These backing stores can be located on different file systems, local to the TeamSite server machine. The functionality that enables multiple backing stores is known as *MultiStore*.

To include MultiStore support in TeamSite (and improve overall performance), a new backing store format needed to be implemented. This format is used by all backing stores created using the current TeamSite release. If you have a backing store created with TeamSite version 4.5.x or

5.0.x, you must convert the old backing store to use the new format as described in either “Converting Backing Stores Using the GUI” on page 242 or “Converting Backing Stores from the Command Line” on page 246. You can also use this procedure to divide your single old-format backing store into multiple new-format backing stores.

Dividing your existing backing store into new multiple stores (possibly on different file systems) enables you to simplify data management, including faster data backup. It also avoids having your backing stores grow to unmanageable sizes.

Note: You can migrate data to your new stores any way you choose, but the data between the stores is completely independent and may not be migrated to other stores using inter-branch copying. Copies remain branch-specific and cannot be used at the backing store level.

Backing stores have a corresponding archive in the VPATH. In previous versions of TeamSite, there was only one archive named `default` with a corresponding backing store called `iw-store\default`. MultiStore functionality allows for multiple backing stores with user-assigned names. Each backing store is similar to the `default` archive in that it contains a single root branch called `main` and is independent of any other store controlled by the server. All mounted backing stores are assigned a unique store ID number and maintain their own set of inodes that are stored persistently inside each backing store.

Backing stores which are named using multibyte characters must be created by editing the `iw.cfg` file. For detailed information, see “Defining Backing Stores in the `iw.cfg` File” on page 252.

Planning the Backing Store Conversion

Before you begin the backing store conversion, read through the conversion overview, and ensure you have satisfied the conversion prerequisites before beginning the actual step-by-step conversion procedure as described in the following sections:

- “Converting Backing Stores Using the GUI” on page 242
- “Converting Backing Stores from the Command Line” on page 246

Also note that there is a *Backing Store Conversion Planning Guide* available on <http://support.interwoven.com> that contains the latest conversion information.

Conversion Overview

This section describes the conversion procedure in very general terms. It is intended to help you understand what is involved in the conversion procedure before you begin.

1. Satisfy the prerequisites (as described on page 241).
2. Decide how you want to organize your new backing store on the target system:
 - Convert your single old-format backing store into a single new-format backing store.
 - Convert your single old-format backing store into multiple new-format backing stores. You can later convert your single new-format backing store into multiple new-format backing stores by using the `iwmigrate` CLT as described on page 259).

Your conversion options are depicted in the graphic at the end of this section.

3. Run `iwfscck -d` on your source backing store to prepare for conversion. The `iwfscck` CLT is described in the *TeamSite Command-Line Tools* manual.
4. Run the `iwconvert` conversion tool from either the GUI or the command line.

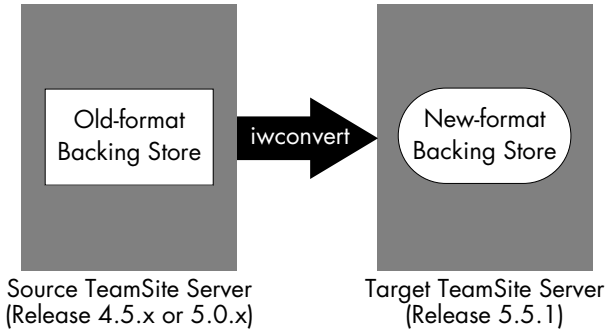
You may repeat this step any number of times depending on what you plan to convert from your existing store, and whether you plan to create multiple new-format stores.

Optional Steps:

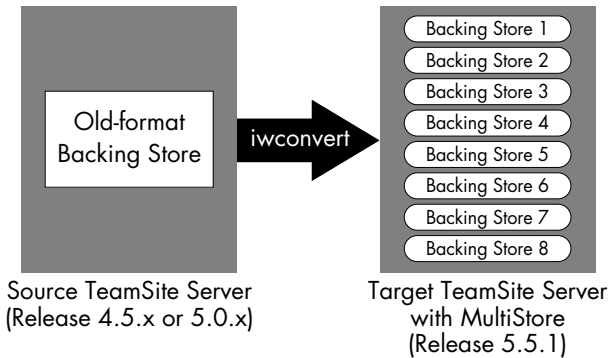
5. Make an edition of your staging area after the initial conversion pass (or passes) is complete.

This edition will contain anything submitted while the initial conversion was running. It should be much smaller than your other editions and should convert faster.
6. Run the `iwfreeze` CLT to prohibit any more changes to the staging area.
7. Create and convert the final edition.
8. If you want to use the source system (where your current TeamSite 4.5.x or 5.0.x installation resides) as your production server *after* the conversion is complete, you will need to install TeamSite 5.5.1 on this server and copy the converted (new-format) backing stores onto this server.

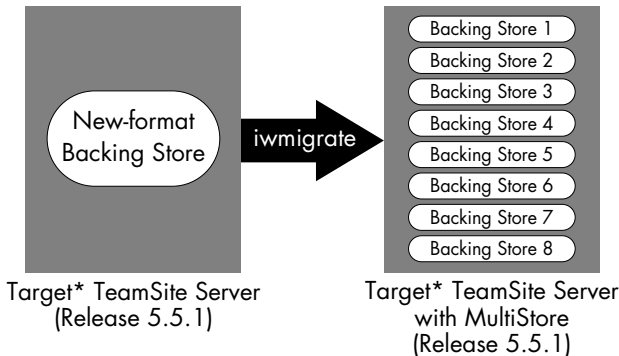
Old-format Single Store → New-format Single Store



Old-format Single Store → New-format Multiple Stores



New-format Single Store → New-format Multiple Stores



* The iwmmigrate procedure can be performed on one TeamSite Server; iwconvert requires two systems.

Conversion Prerequisites and Tips

- You must have two systems running the same operating system (they do not need to be the same version) and be located on the same network. This document refers to these as the *source* system (which contains your current, old-format backing store) and the *target* system (which will host the new-format version of the backing store).
- Ensure that TeamSite 4.5.x, or 5.0.x is installed and licensed on the source system.
- Back up your existing installation and backing store.
- Ensure that TeamSite 5.5.1 is installed and licensed on the target system.
- The target system must have at least as much disk space as the source system.
- Consider creating separate backing stores for branches that meet the following criteria:
 - Distinct deployment targets
 - Legacy or infrequently accessed
 - Distinct ownership within your organization
- Decide how you are going to divide the source backing store for conversion. Depending on the size of your source backing store, and the organization of your TeamSite implementation, you may choose to convert a range of editions, a single edition, or branch-by-branch.
- Before running the `iwconvert` command, run `iwfsck -d` on your source backing store to prepare for conversion. The `iwfsck` CLT is described in the *TeamSite Command-Line Tools* manual.
- Allow plenty of time for the conversion to complete. While there are many variables affecting the time it takes to convert a backing store, tests show that the conversion runs at approximately 500 megabytes (MB) per hour. Note that this is a very general number and you should not be concerned if your conversion runs at a different rate.
- The user who mounts `iw-store` and `iwserver` on the source machine, must be a domain user who is in the Administrators group on both the source and target machines.
- Plan to publish new editions of the staging area if you allow users to submit files while the conversion is running. You may have to do this multiple times. Eventually, you will need to freeze your source server to prohibit users from using TeamSite during the final conversion.
- You can use the `iwmigrate` CLT to migrate data between new backing stores. You can also use your operating system's copy functionality, but you will lose history and version information.
- You can have a maximum of eight active backing stores on your target system. You can have more than eight backing stores, but only eight can be active.

- You can create backing stores on NFS-mounted remote servers if you are using disk management devices.
- Workflow tasks must be completed prior to conversion.
- To avoid having the backing store conversion procedure create an unknown user on the target system, the user doing the conversion must have a user account with access to every file and ACL on both systems. If an unknown user is created, you can use the `iwidmap` CLT (as described on page 38) to remap the unknown user to an appropriate user on the target system.
- Your backing store represents a major investment to your organization. If for any reason you feel that you want help with the conversion process, please contact Interwoven Client Services (<http://www.interwoven.com/services>).

Converting Backing Stores Using the GUI

Whether you use the conversion GUI or convert from the command line, the backing store conversion is done by the `iwconvert` program. This utility can be run directly from the command line (as described in “Converting Backing Stores from the Command Line” on page 246), from a script generated by the conversion GUI, or interactively by using the conversion GUI. The bulk of the conversion is done while your existing TeamSite server is running, but you will need to freeze your source server (or prohibit users from using TeamSite) for the final few editions to ensure that no data is lost.

The backing store conversion GUI is a CGI program (`iwconvert.cgi`) that is installed by the TeamSite installation program and displayed in your Web browser. It is supported by a process called `iwconvertserver` that communicates with remote TeamSite servers and invokes `iwconvert` on behalf of the GUI. You must run the `iwconvertserver` process manually for the GUI to function properly.

Complete the following procedure to convert your old-format backing store to use the new backing store format:

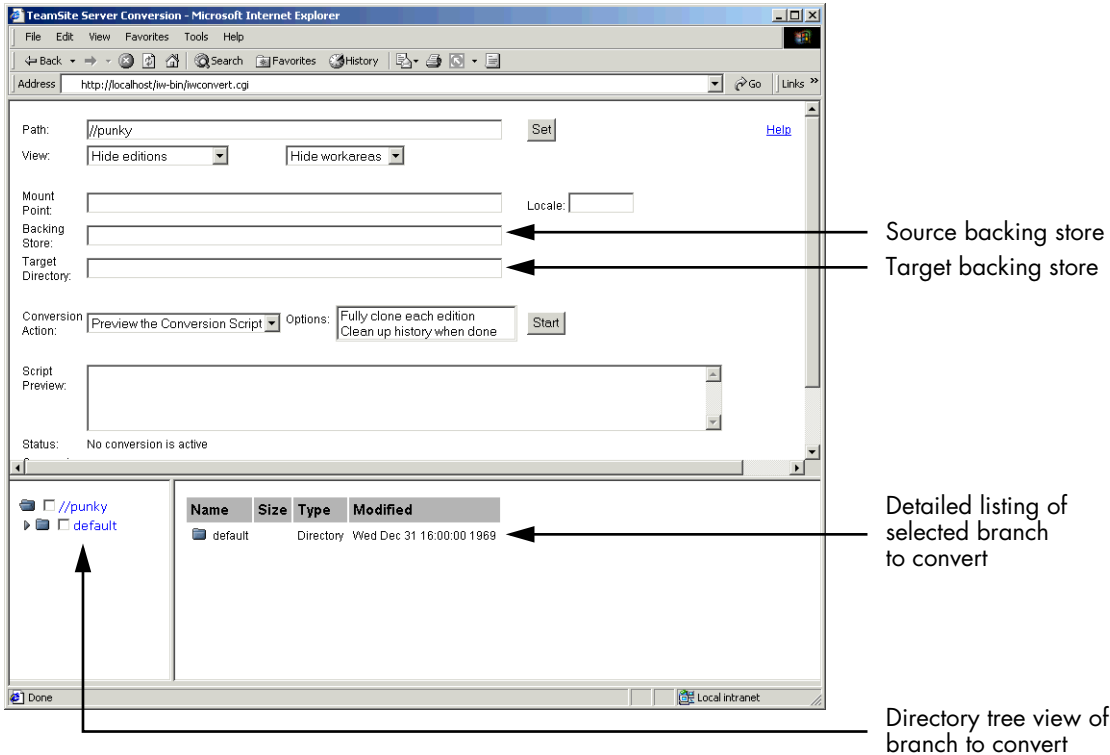
1. Log into the TeamSite 5.5.1 system as Administrator.
2. Run the `iwconvertserver` utility directly from the command line in `iw-home\bin`, for example:

iwconvertserver -S

- 3. Open your Web browser.
- 4. Type **http://localhost/iw-bin/iwconvert.cgi** in the **Address** field (Internet Explorer) or the **Location** field (Netscape).

Note: If you are not working on the same computer that will contain the new backing store, you must specify the system name instead of localhost in this step. For example, if you wanted the new backing store to be located on a server named factotum, type: **http://factotum/iw-bin/iwconvert.cgi**

The conversion GUI is displayed in your default browser:





5. In the **Path** field, enter the network version path (vpath) to the TeamSite server containing the source backing store.

If the source backing store is located on a remote server you must begin the path with two forward slashes (//). For example, in the previous graphic, the old-format backing store to be converted is located on a remote server named punky.

6. Click **Set** to display the source tree of the vpath of the source backing store.
7. Set the **View** menus to filter the files that are displayed in the directory tree view of the source backing store (lower left frame in the conversion GUI). The options are:
 - **Hide editions**—Individual editions are not displayed. Converting this branch converts all corresponding editions.
 - **Show editions**—Displays individual editions. You must select which editions are converted. (Expanding an EDITIONS directory containing a large number of editions can be very slow.)
 - **Show recent editions**—Displays the 10 most recent editions. You must select which editions are converted.
 - **Hide workareas**—Workareas associated with the selected branch are not displayed or converted.
 - **Show workareas**—Individual workareas are shown. You must select which workareas are converted. (Expanding a WORKAREAS directory containing a large number of workareas can be slow.)
8. Click **Set** to update the source tree if you modified the **View** menus (in step 7).
9. In the **Mount Point** field, enter the file system path to the TeamSite virtual file system of the source machine, for example, F : / .
10. In the **Locale** field, enter the locale of the backing store to be converted, or leave the field blank to use the default locale.

This entry is used to specify how non-ASCII metadata is interpreted. The following are valid locales:

- iso-8859-1 • iso-8859-2 • iso-8859-3 • iso-8859-4 • iso-8859-5
- iso-8859-6 • iso-8859-7 • iso-8859-8 • iso-8859-9 • iso-8859-15

- euc-jp • euc-tw • euc-cn • euc-kr • shift_jis
- big5 • gb2312 • utf-8 • utf8

11. In the **Backing Store** field, enter the file system path to the source backing store, for example: `//punky/iw-store/default`.

This field is optional, but providing the path results in a significantly faster conversion.

12. In the **Target Directory** field, enter the name of the directory where the new backing store is to be created, for example: `C:/iw-store/newStore1`.

13. Select one of the following actions from the **Conversion Action** menu:

- **Preview the Conversion Script**—Displays the `iwconvert` command that is generated based on the selected branches and options in the Script Preview field when **Start** is clicked.
- **Generate the Conversion Script**—Displays the `iwconvert` command that is generated based on the selected branches and options in the Script Preview field, and writes the command to a script file when **Start** is clicked.

The script files use the `iwconvert_##.sh` naming convention (where `#` represents an integer) and is created in the `C:/iw-home/local/logs` directory on the target system.

- **Run Conversion**—Displays the `iwconvert` command that is generated based on the selected branches and options in the Script Preview field, writes the command to a script file, and invokes the `iwconvert` command when **Start** is clicked.

14. Select none, one, or both of the following `iwconvert` options (use Shift+click to select both options):

- **Fully clone each edition**—Runs `iwconvert` with the `-f` option, so that editions are cloned without any submit event history. Use this option if there are gaps in the set of editions that you are converting, or if you do not want to save the submit history.
- **Clean up history when done**—Runs `iwconvert` with the `-c` option to generate submit events correctly.

If you do not select one or both of the options, all history is copied.

15. Click **Start** to initiate the action defined in step 13 and step 14 and to display the status of the conversion:

- **Status**—Displays whether or not `iwconvert` is currently running from the GUI. If “`iwconvertserver not enabled`” is displayed, you must run the `iwconvertserver` process that supports the GUI (described in “Administration CLTs” on page 257).



- **Conversion Step**—Displays the current `iwconvert` command line if it has been started from the GUI.
 - **Conversion Detail**—Displays the progress of the conversion if `iwconvert` has been started from the GUI.
16. If files are submitted during the conversion procedure, you will need to freeze the server (by using `iwfreeze`), create a new edition that contains these files, and repeat the conversion procedure.

Note: The `iwconvert` program creates temporary workareas (`temp_workarea`) in each converted branch as a by-product of the conversion process. You should manually delete these after the conversion.

Converting Backing Stores from the Command Line

The TeamSite installation program installs a set of command-line tools in the `iw-home\bin` directory. All of these tools are documented in the *TeamSite Command Line Tools* manual that corresponds with your platform. For convenience, the new CLTs are also included in this document. The `iwconvert` CLT is described in this section the other new CLTs are described in “Administration CLTs” on page 257.

`iwconvert.exe` Command-Line Tool

The `iwconvert` CLT converts old-format (TeamSite 4.5.x and 5.0.x) backing stores to the new high-performance backing store format.

Complete the following steps to optimize the conversion process.

- Upgrade your source machine to TeamSite 4.5.1 Service Pack 2, or TeamSite 5.0.1 Service Pack 2 (or higher), with all available patches.
- Ensure you have the most recent version of the `iwconvert` tool.

Updates to the `iwconvert` and `iwmigrate` tools shipped with TeamSite 5.5.1 will be available on the Interwoven support website. Before using either CLT, check the Interwoven support website (<http://support.interwoven.com>) to ensure you have the most recent version of each tool.

- Before running the `iwconvert` command, run `iwfsck -d` on your source backing store to prepare for conversion. The `iwfsck` CLT is described in the *TeamSite Command-Line Tools* manual.

Options

The following options are valid for the `iwconvert` command:

<code>-h</code>	Displays the usage message.
<code>-v</code>	Displays the version number.
<code>-b <i>branch_vpath</i></code>	Location of the branch that contains the editions or workareas to be converted. If the <code>vpath</code> begins with <code>//hostname/</code> the branch is located on a remote TeamSite server. Note: Branches are converted recursively—all editions in subbranches under the specified branch are also converted unless the <code>-d</code> option is specified.
<code>-c</code>	Cleans up the history information of a previously converted backing store. Requires that <code>-o</code> is also specified. This <code>iwconvert</code> step must be performed last, as a separate step, because it may have interbranch dependencies. Note that this action may safely be executed multiple times.
<code>-d</code>	Do <i>not</i> recursively convert subbranches.
<code>-s <i>starting_edition</i></code>	Specifies the first edition in a range of editions to be converted for the specified branch (the default is <code>INITIAL</code>). Requires that <code>-b</code> , <code>-d</code> , <code>-o</code> , and <code>-m</code> are also specified.
<code>-e <i>ending_edition</i></code>	Specifies the last edition in a range of editions to be converted for the specified branch (the default is the most recent edition). Requires that <code>-b</code> , <code>-d</code> , <code>-o</code> , and <code>-m</code> are also specified.



-f	<p>Forces a full clone of each edition <i>without</i> the history of submit events for the editions.</p> <p>Use this option if there are gaps in the set of editions that you are converting, or if you do not want to save the submit history.</p>
-l locale	<p>Specifies the native locale of the backing store being converted and how non-ASCII metadata is interpreted. If this option is not specified it defaults to LC_LOCAL.</p>
-m <i>iwmnt_mount_point</i>	<p>Specifies the mount point for the existing (source) <i>iwserver</i> installation. Required with <i>-b</i>, <i>-r</i>, and <i>-w</i>.</p>
-n <i>old_backing_location</i>	<p>Use direct access to the old backing store for faster conversions. Can be used with <i>-b</i>, <i>-r</i>, or <i>-w</i>.</p> <p>If this option is not specified, <i>iwconvert</i> will run slowly due to calls to <i>sci_GetPredecessors()</i>.</p>
-o <i>new_backing_location</i>	<p>Location of the new backing store. This must be a path to the store root. For example, the store named <i>default</i> is specified by:</p> <p><i>/local/iw-store/default</i>.</p>
-w <i>workarea_name</i>	<p>Converts the specified workarea. Requires that <i>-b</i>, <i>-o</i>, and <i>-m</i> are also specified.</p>
-x	<p>Increases the verbosity level. Maximum verbosity is level 3, expressed as <i>-x -x -x</i>.</p>
Ctrl+c	<p>Stops <i>iwconvert</i> at the end of the edition currently being converted. When you restart the conversion, <i>iwconvert</i> ignores the editions in the branch that have already been converted and converts the remaining editions.</p>

Usage Summary

- Convert editions:

```
iwconvert -o new_backing_store_location -m iwmnt_mount_point  
[-n old_backing_store_location] -b branch_vpath [-d [-s  
starting_edition] [-e ending_edition] ]
```


- Convert a workarea:
`iwconvert -o new_backing_store_location -m iwmnt_mount_point
[-n old_backing_store_location] -b branch_vpath -w workarea_name`
- Clean up history:
`iwconvert -o new_backing_store_location -c`

Example

```
iwconvert -m f:/ -o d:/iw-store/Safari -n C:\iw-store\default  
-b //bgunn/default/main/www
```

Conversion Procedure

Convert each branch (and its associated subbranches) by completing the following procedure.

1. Map a network drive to the source backing store (typically `c:/iw-store/default`) from the target server.
2. Map a network drive to the source server (`\\hostname\iwserver`).
3. Decide which editions are to be converted.

Branches are converted recursively—all editions in the subbranches under the specified branch are also converted unless the `-d` option is specified. You can convert an individual edition (typically the most recent) or, if the `-d` option is specified, a range of editions.

For example, the range of `INITIAL` to `ed_0006` would convert seven editions: `INITIAL` and `ed_0001` through `ed_0006`. Each range of editions converted requires a separate invocation of `iwconvert`.

4. If files have been submitted to the staging area since the last edition was published, publish a new edition.
5. Issue the `iwconvert` command from the `iw-home\bin` directory:

```
iwconvert -o new_backing_store_location -m /mount_location -b  
source_branch -d -s start_of_edition_range -e end_of_edition_range
```

For example, using the example edition range from step 3, on a remote server named `factotum`, and a branch named `default/main/intranet`:



```
iwconvert -o c:\iw-store\default -m F:\ -b  
\\factotum\default\main\intranet -d -s INITIAL -e ed_0006
```

Note: You should save the `stdout` and `stderr` output from the `iwconvert` procedure to a log file by appending the following to the command in this step:
...INITIAL -e ed_0006

6. Convert the changes that occurred while the conversion was running by performing either of these steps:

- Submit all changes from workareas to staging, then publish a new edition that contains these changes. Convert this new edition as described in step 5.
- Convert again using the `-w` option to convert the workareas that contain changes not submitted to the staging area before the conversion described in step 5.

```
iwconvert -o c:\iw-store\default -m F:\ -b  
\\factotum\default\main\intranet -w jerome
```

If you have changes in a large number of workareas it is easier to have users submit their changes and publish and convert a new edition rather than converting the workareas that contain changes.

This step should complete much faster than your original conversion.

7. Freeze your source server by running the `iwfreeze` command from the `iw-home/bin` directory specifying a large number of seconds for the freeze, for example:

```
iwfreeze +50,000
```

8. Repeat step 6 to convert the final changes that were made during the second conversion.
9. Run `iwconvert` with the `-c` option to clean up the second-predecessor links in the new-format (target) backing store, for example:

```
iwconvert -o c:\iw-store\default -c
```

These links are created by TeamSite operations including Copy To. The clean up of history must be done as a separate pass at the end of other `iwconvert` passes because the second-predecessor links can point in various directions between branches and areas in the backing store, and the referenced objects may not be converted at the time they are needed.

If the workareas contain versions of files that have not been converted, the correct contents of those files are copied into the workarea, and those files will show up as modified.

Note the following:

- The `iwconvert` program creates temporary workareas (`temp_workarea`) in each converted branch as a by-product of the conversion process. You should manually delete these after the conversion.
- Do not use the Registry Editor to point to the newly converted backing store. While you can move your default store by editing the registry, this method is not reliable and only works when the directory is named `default`. If you use this method and convert to a destination directory with a subdirectory named something other than `default`, the new store will not be activated.

Creating Multiple Backing Stores

Multiple backing stores can be created using two different methods depending on where you want to locate them, and whether you want to use multibyte characters in their names.

- `iwstoreadm CLT`—Creates and activates new backing stores when issued with the `-a` option.
 - Accepts ASCII characters for store names.
 - Creates the new backing store in the default location (typically `C:\iw-store\`).
 - Does *not* allow for a descriptive comment to be added to the backing store.
- Editing the `iw.cfg` file—Defines backing stores with entries in the `[iwserver]` section of the `iw.cfg` file.
 - Accepts multibyte characters for the store name (though the path to the store must use ASCII characters)
 - Creates the new backing store in any location.
 - Allows you to add a descriptive comment to the backing store. This comment is displayed when the active backing stores are listed from the command line, or displayed in the TeamSite GUIs.
 - Must be activated by using the `iwstoreadm CLT` with the `-a` option.



If you want to define backing stores by editing the `iw.cfg` file, complete the procedure described in the next section. If you want to create backing stores using the `iwstoreadm` CLT, complete the procedure described in “Creating Backing Stores Using the `iwstoreadm` CLT” on page 255.

Defining Backing Stores in the `iw.cfg` File

As previously mentioned, the advantages of defining backing stores in the `iw.cfg` file include the ability to use multibyte characters in store names and to locate the backing store in a directory other than `C:\iw-store\`.

User-defined backing stores which are named using multibyte characters, must have a corresponding entry in the `iw.cfg` file. While the name of the backing store can be defined in multibyte characters, the backing store location *must* be defined using ASCII characters. All backing store data is stored in UTF-8 encoding.

Complete the following procedure to create backing stores defined in the `iw.cfg` file:

1. Ensure that the user you are logged in as has an entry in the Master role file (`iw-home\conf\roles\master.uid`).
2. Open the `iw.cfg` file in a text editor.
By default, the `iw.cfg` file is located in `iw-home\etc`.
3. If you are using multibyte characters for the store name, specify the encoding of your `iw.cfg` file by creating the following entry as the first line in the file—it *must* be the first line or it will be ignored.

```
[iwcfg]
encoding=locale_name
```

where *locale_name* is one of the following locales:

- `shift-jis` (Japanese)
- `cp1252` (French or German)

For example:

```
[iwcfg]
encoding=shift-jis
```

Note: The locale entry must match the encoding of your text editor. Refer to page 314 for details about text editor encodings.

4. Append the following entry to the [iwserver] section to define additional backing stores:

```
store_directory_store_name=absolute_path_to_backing_store
```

For example:

```
store_directory_salesAsia=C:\salesAsia
```

Note: The *absolute_path_to_backing_store* must be in ASCII while the *store_name* and the optional *descriptive_comment* (described in step 5) can be in high-ASCII or multibyte characters.

5. Optionally, add a comment to the [iwserver] section below the backing store you just defined:

```
store_comment_store_name=descriptive_comment
```

For example:

```
store_comment_salesAsia=Store for Demo
```

The completed entry, should look like this:

```
[iwserver]
existing iwserver entries
store_directory_salesAsia=C:\salesAsia
store_comment_salesAsia=Store for Demo
```

6. Save and close the iw.cfg file.
7. Run the iwreset CLT to have the TeamSite server read the changes to the iw.cfg file.
8. Run the iwstoreadm CLT with the -a option to create the newly defined backing store:

```
>iwstoreadm -a salesAsia
```

The iwstoreadm CLT checks the iw.cfg file to see if a *store_directory* or *store_comment* entry exists, when it finds these entries, their definitions are used to create the backing store.

The system then activates and mounts the new backing store.

9. Run the `iwstoreadm` CLT with the `-l` option to list all active backing stores:

```
>iwstoreadm -l
```

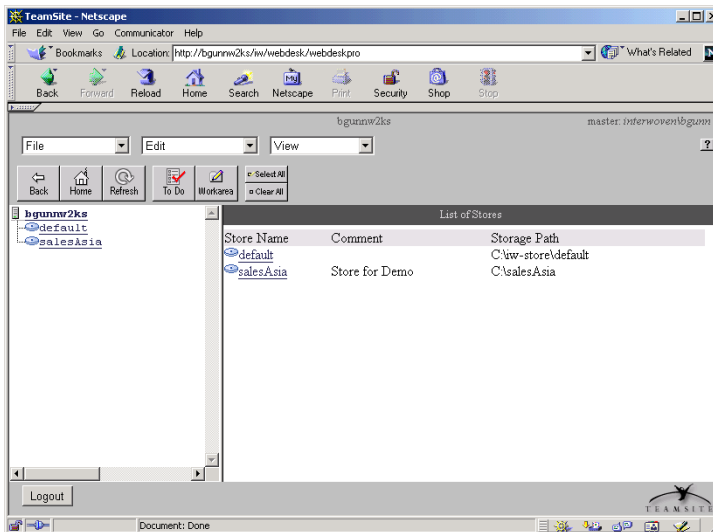
The system displays the following:

<u>Name</u>	<u>Store_Directory</u>	<u>ID</u>	<u>Comment</u>
default	C:\local\iw-store\default	0x64	
salesAsia	C:\salesAsia	0x65	Store for Demo

10. Open your web browser and log in to WebDesk Pro or WebDesk.

11. Click **Workarea** (WebDesk Pro) or the **Files** tab (WebDesk).

The backing store and comment you created is listed in the GUI.



Notes:

- You can repeat the procedure to create any number of backing stores, but you can only have eight active at one time.
- You can edit the `store_directory_storename` entries to move backing stores defined in `iw.cfg`.

Creating Backing Stores Using the iwstoreadm CLT

The following procedure describes the creation of backing stores from the command line using `iwstoreadm`. It also describes viewing the newly created backing stores in both the command window and the TeamSite WebDesk Pro interface.

- 1. Ensure that the user you are logged in as has an entry in the Master role file (*iw-home\conf\roles\master.uid*).
- 2. Issue the `iwstoreadm -a store_name` command to create a new store, for example:

```
>iwstoreadm -a store1
```

store1 is created in `C:\iw-store\` and activated.

- 3. Type the following command to list the active backing stores:

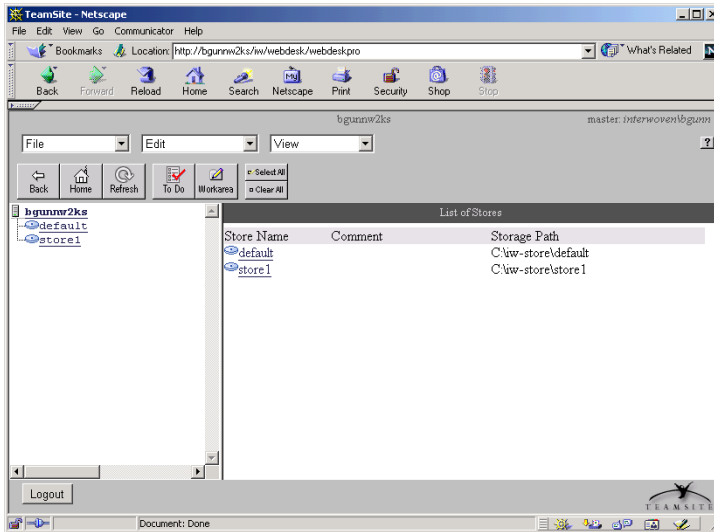
```
>iwstoreadm -l
```

The following listing is displayed:

Name	Store Directory	ID	Comment
----	-----	--	-----
default	C:\iw-store\default	0x64	
store1	C:\iw-store\store1	0x65	

- 4. Open your web browser and log in to WebDesk Pro or WebDesk.
- 5. Click **Workarea** (WebDesk Pro) or the **Files** tab (WebDesk).

The backing store you created is listed in the GUI.



6. Click store1.

Note that all backing stores (including the system-generated default) contain a Main branch, and a STAGING area that is based on an INITIAL edition.

Administration CLTs

This section describes the new command-line administration utilities designed for use with the conversion GUI and new backing store functionality. The conversion CLT (*iwconvert*) is described on page 246.

iwstoreadm.exe

Backing store administration involves creating, activating, and deactivating backing stores by using the *iwstoreadm* CLT. When the *iwstoreadm -a storename* command is issued, the following sequence is triggered:

- The *iw.cfg* file is checked to see if a *store_directory* or *store_comment* entry exists, if it does, their definitions are used to create the backing store. If these entries do not exist:
- The backing store directory is automatically created in *C:\iw-store* and populated with a minimal backing store containing a branch named *main*. The store remains active until explicitly deactivated by using *iwstoreadm -d* (even if the server is stopped and restarted).

Deactivating a store does *not* delete it. A store can be deactivated, moved to a new location, and reactivated using original name though it will be assigned a new store ID.

iwstoreadm.exe Options

The following options are valid for the *iwstoreadm* command:

<i>-a store_name</i>	Activates an existing store, or creates and activates a new store. You must have an entry in the Master role file (<i>iw-home\conf\roles\master.uid</i>) to execute this option.
<i>-d store_name</i>	Deactivates an existing store. You must have an entry in the Master role file (<i>iw-home\conf\roles\master.uid</i>) to execute this option.
<i>-h</i>	Displays the usage message.
<i>-l</i>	Lists active stores.

Usage

```
iwstoreadm [-l] [-a store_name] [-d store_name]
```

Example

```
>iwstoreadm -l
```

Displays the active backing stores:

Name	Store Directory	ID	Comment
----	-----	--	-----
default	C:\iw-store\default	0x64	
store2	C:\iw-store\store2	0x65	

iwidmap.exe

A command line tool called `iwidmap` is included to change the mapping between the SID and the token. It can also be used to refresh the mapping when the same names are used, and the SID has changed. For more information about SID mapping, see “SID Changes to the TeamSite Backing Store” on page 260.

Usage

```
iwidmap [-v] [-h] (-u | -g) [-a][-c <user1> <user2>] [-x <file> | -i <file>]  
backing-store
```

-v	Displays the version of this program.
-h	Displays the usage message.
-u	Update userid mapping.
-g	Update groupid mapping.
-a	Update all entries in the ID map.
-c <user> <user2>	Update user1 to user2.
-x <file>	Extract to file.
-i <file>	Import from file.
backing-store	Location of the backing store.

Example

```
iwidmap -u -c jgarcia rhunter c:\iw-store\NewReleases
```

iwmigrate.exe

The `iwmigrate` CLT is similar to `iwconvert` except that it accepts new-format backing stores as its source. It can be used to split a single new-format backing store into multiple backing stores, or to move the contents of a store to another location without losing the history of submit events for the editions.

Note: Updates to the `iwconvert` and `iwmigrate` tools shipped with TeamSite 5.5.1 will be available on the Interwoven support website. Before using either CLT, check the Interwoven support website (<http://support.interwoven.com>) to ensure you have the most recent version of each tool.

Usage

```
iwmigrate [-h] [-v] [-x] [-m mount_location] -o new_backing_location
[-b branch_vpath] [-s starting_ed] [-e ending_ed] [-n old_backing_location]
[-w workarea_name] [-c] [-d] [-f] [-l]
```

-h	Display this message.
-v	Display version number.
-b branch	Specify source branch for migration.
-x	Increase verbosity level. Maximum verbosity is level 3, expressed as -x -x -x.
-m mount_location	Specify mount location of backing store, for example: F:\
-o new_backing_location	Specify new backing store location.
-n old_backing_location	Specify old backing store location.
-d	Do <i>not</i> recursively convert subbranches
-s starting_ed	Specify starting edition for migration. Default is the INITIAL edition (this option can only be used with -d)
-e ending_ed	Specify ending edition for migration. Default is the most recent edition.
-f	Full clone of every edition (does <i>not</i> preserve history).



- | | |
|-----------|---|
| -r | Clean up history information (use on the final pass). |
| -l locale | Specify the native locale of the backing store being migrated (if different from LC_LOCAL for this system). |

Example

```
iwmigrate -m F:\Safari -o d:\iw-store\safari_on_line -b
```

iwconvertserver.exe

The `iwconvertserver` process supports the conversion GUI by communicating with remote TeamSite servers and invoking `iwconvert` on behalf of the GUI. You must run the `iwconvertserver` process manually for the GUI to function properly.

1. Change to the `iw-home/bin` directory, for example:

```
>cd C:\Program Files\Interwoven\TeamSite\bin
```
2. Either:
 - Run the `iwconvertserver` utility directly from the command line:

```
iwconvertserver -S
```
 - Install it as a service, and then start the service using the Service Control Manager:

```
iwconvertserver -is
```

SID Changes to the TeamSite Backing Store

Previous releases of TeamSite have stored Windows NT security identifiers (SID) representing users and groups directly in the backing store. This would cause problems when converting the backing store onto different systems.

The new-format backing store uses a unique 32-bit ID generated by the TeamSite server instead of storing the SID. A one-to-one persistent mapping exists between the TeamSite generated ID and the SID. Whenever the SID has to be written out to the backing store, the mapping is checked to obtain the TeamSite ID, which is substituted for the SID. When a restore is attempted, the reverse lookup takes place and the appropriate SID is recovered.

A command-line tool called `iwidmap` is included to change the mapping between the SID and the token. It can also be used to refresh the mapping when the same names are used, but the SID has changed. Details about the `iwidmap` CLT are included on page 258.

Chapter 9

Backing Up TeamSite

The TeamSite backing store represents a tremendous investment in resources and is a valuable corporate asset. As such, it should be backed up daily, or even more frequently, to minimize the possibility of damaged or lost data. TeamSite 5.5.1 requires the use of third-party backup solutions. Any backup mechanism that can guarantee exact time and state directory content recovery can be used.

Integrating with Third-Party Backup Solutions

Interwoven recommends using a high quality third party backup solution for protecting the backing store data. When evaluating a backup solution, the following criteria are essential:

- The backup method must provide a way to perform an `iwfreeze` operation prior to performing the backup. This must be done to assure that the backing store does not change during the backup. The backup method must then perform an `iwfreeze --` operation to allow writes to the backing store when the backup is finished.
- The backup method must be fast enough to perform a full or incremental backup of the backing store within a reasonable length of time. The maximum allowable length of time depends on the requirements of the particular installation, but should probably be less than 12 hours.
- The restore method must provide a way to do a complete state-restore of a directory as of a given time. This means that when a directory is recovered, the contents must match exactly what was in the directory at the time the backup was performed. Only files that were present at the time of the backup must be present in the restore. That is, if a file was deleted from the

original directory between backups, it should not be present in the restore. Some backup and restore products regard all backed-up files to be “sticky,” i.e., as long as a file ever existed, it will be present in the restoration regardless of whether it was deleted prior to the last backup.

Additional criteria to consider are:

- An automated backup execution facility capable of performing full backups followed by level (preferred) or incremental backups to provide a customizable backup strategy.
- Automated backup media management and manipulation (for example, a tape jukebox or silo).
- The ability to make copies of completed backups for offsite storage.

If the available backup method is efficient and inexpensive (compared to the value of the data being protected), the TeamSite workareas can also be backed up to allow users to recover individual files or directories from their workareas, rather than having to recover the entire backing store. This is a very convenient feature for users, but can come at a relatively high price in terms of extra time and space needed for these redundant backups. Although the virtual files which comprise much of TeamSite’s file system mount (Y:) take up no extra space on the TeamSite server, if the actual TeamSite workareas are backed up, the virtual files in the workareas will be treated as actual files and will take up space in the backup media.

It is not absolutely necessary to freeze the TeamSite backing store while you are backing up workareas; however, failure to freeze the backing store while you are backing up the backing store itself can result in possible data loss and corruption.

Note: If you are using multiple backing stores, you can back up each store independently. The `iw-store` directory should be backed up if you have in-progress workflows or batch jobs that you do not want to lose. You can freeze and unfreeze the workflow store just like any other store, but you cannot move it outside of `iw-store`.

Backing up workareas alone is not a substitute for backing up the TeamSite backing store. If you only back up the files that appear in the TeamSite file system mount, you will lose important metadata such as version histories and file status. Always back up the actual TeamSite backing store whether or not you back up individual workareas.

Suggested Strategies for Incremental Backups

It is possible to implement a “level-oriented” backup if a sufficiently sophisticated backup solution is available. For example, a full backup can be performed on the first Saturday of the month, then incremental backups that build on each other can be performed for the rest of the week. On the second Saturday of the month, a “super-incremental” backup based on the original full backup done on the first Saturday is performed. The super-incremental backup supersedes all of the previous incremental backups. Only the first full backup and super-incremental are needed to completely recover the backing store. For the subsequent week, incremental backups are again performed based on the super-incremental backup done on the second Saturday. The following Saturday, another super-incremental backup based on the previous super-incremental file is performed, again eliminating the need for the previous week’s incrementals to recreate the backing store. To perform a recovery at this point, restore the original full backup, then each super-incremental in sequence, and finally the balance (if any) of the current week’s incrementals.

This tiered, or level-oriented backup can be repeated on a monthly basis to produce a week-by-week archive of the backing store. To reproduce the backing store as of any particular Saturday, recover the full backup from the beginning of the month, then apply each Saturday backup in turn until the desired Saturday is reached.

To determine your optimal backup strategy, you must analyze the tradeoffs of convenience and speed in backing up versus simplicity and speed of restoration, and decide what best suits your needs. A strategy using a single full backup and an indefinite string of incrementals is optimized for backup speed, but the amount of time required to perform a full recover of the backing store grows with each passing day as a new incremental is added to the list. Every backup must be preserved to be able to recover the backing store. One benefit of this method is that a complete daily archive of the backing store will be preserved.

The opposite extreme is to perform a full backup every day. Each backup will take the maximum amount of time to perform, but only one recover needs to be done to completely recreate the backing store. If you only preserve the previous day’s backup, no history of the backing store will be retained, but the amount of storage space used by the backups is minimized.

Appendix A

TeamSite Configuration Files

The following files contain information about your TeamSite server configuration:

Configuration File	Function
<i>iw-home</i> \etc\iw.cfg (default location—see “Location of iw.cfg” on page 268 for more information)	Contains various parameters necessary for the operation of TeamSite, as described in the Chapter 5, “Configuring the TeamSite Server.”
<i>iw-home</i> \local\config\submit.cfg	Specifies all file permissions that will automatically be changed at submit time.
<i>iw-home</i> \local\config\autopprivate.cfg	Specifies what types of files will automatically be marked private.
<i>iw-home</i> \local\config\file_encoding.cfg	Contains rules that determine the character encoding of the contents of files that do not specify their encoding. See page 269 for information about creating these rules.
<i>iw-home</i> \local\config\iwtemplates.cfg	Specifies which New File templates will be used in which TeamSite areas.
<i>iw-home</i> \conf\roles\master.uid	Contains a list of all users who can log in as a Master user.
<i>iw-home</i> \conf\roles\admin.uid	Contains a list of all users who can log in as an Administrator.
<i>iw-home</i> \conf\roles\editor.uid	Contains a list of all users who can log in as an Editor.
<i>iw-home</i> \conf\roles\author.uid	Contains a list of all users who can log in as an Author.

The locations of most of these files can be changed (see See “File Locations” on page 146.).

Location of iw.cfg

If `iw.cfg` does not exist in the default location, TeamSite will look for it in the following locations, in order:

`iw-home\local\etc\iw.cfg`

`iw-home\etc\iw.cfg`

the Registry key `HKEY_LOCAL_MACHINE\Software\Interwoven\TeamSite\iw-config`

If `iw.cfg` is not found in any of these places, TeamSite will assume the default values for `iw.cfg` settings.

Location of Roles Files

TeamSite looks for the roles files (`author.uid`, `editor.uid`, `admin.uid`, `master.uid`) in the following locations, in order:

The value in the `iwroles` field in the `[locations]` section of `iw.cfg`, if it exists.

`iw-home\conf\roles`

`iw-home\local\config\roles`

`iw-home\config\roles`

`iw-home\local\config\roles`

Appendix B

Specifying Content Encoding

TeamSite 5.5.1 introduces a configuration file called `file_encoding.cfg` that enables you to create rules to determine the character encoding of the contents of files that do not specify their encoding. The `file_encoding.cfg` file (located by default in *iw-home\local\config*) uses an XML-based language called `regex_map`. The `regex_map` format is designed to be structured enough for maintainability, and extensible so that the same format may be used in future configuration files. This file contains a `<regex_map>` element, which contains rules to map `vpaths` (directory paths) to the character encoding specification of file contents.

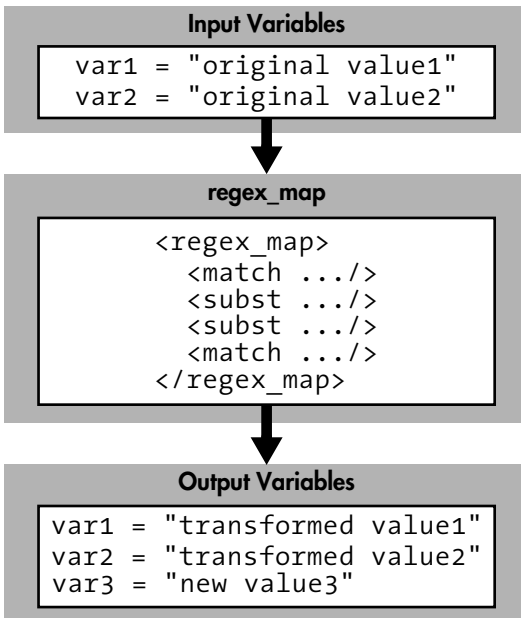
For TeamSite to correctly interpret a text document, it is necessary to know the character encoding in which its contents are represented. Unlike an HTML document that can declare the encoding of its contents using an `<HTTP META="Content-Type" CONTENT="text/html; charset=charsetname">` tag, a plain text file has no mechanism for storing this information. The encoding is required for certain TeamSite functionality including SmartContext Editing (SCE) and the Source Differencing and Interwoven Merge tools.

In previous releases, SCE relied on the Content-Type header from the content webserver to specify the encoding of plain text files. This required you to configure the encoding at your content webserver which may limit flexibility and scalability. By default, the Source Differencing and Interwoven Merge tools assumed that text files are encoded in ISO-8859-1, which is not suitable for content in eastern Asian languages.

The sections that follow describe the `regex_map` language, and how it is used to specify the character encodings of text files used by TeamSite.

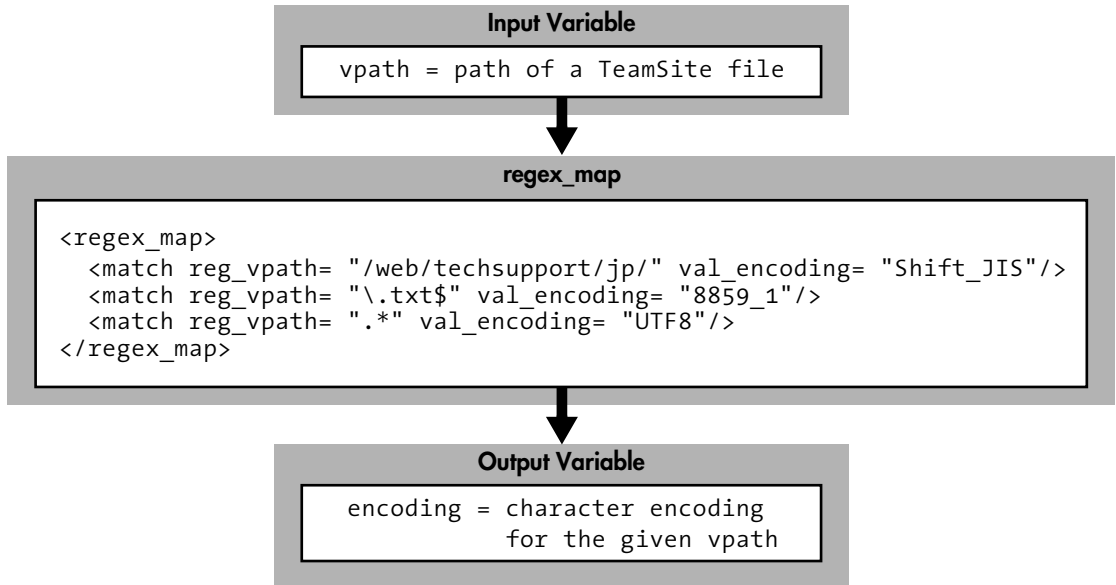
regex_map Defined

A `regex_map` is a filter that transforms a set of input variable values into a set of output variable values through a set of rules written in XML using the following form:



Simple regex_map Example

The following `regex_map` determines the character encoding of TeamSite files. Each `reg_vpath` means that a match is to be performed on the `vpath` variable, and each `val_encoding` assigns a result if the match succeeded.



In the preceding `regex_map` example:

- If the input `vpath` variable is `"/x/y/z.txt"`, the resulting encoding variable is set to `"8859_1"` because `"/x/y/z.txt"` ends with `".txt"`.
- All files in the `/web/techsupport/jp` branch are encoded in Shift-JIS, because their `vpath` begins with `"/web/techsupport/jp/"`.
- If the input `vpath` variable is anything other than `"/web/techsupport/jp/"`, the output encoding variable is set to `"UTF8"` because `".*"` matches any string.

Note that each rule within `<regex_map>` is evaluated in order, and that the first `<match>` tag with a regular expression that matches the input variable `vpath` is used and subsequent rules are ignored. Therefore, it is important for the `<match reg_vpath= ".*" val_encoding= "UTF8"/>` rule to appear last.

The regex_map Format

A `regex_map` consists of a `<regex_map>` element that contains substitution and match rules expressed by using `<subst>` and `<match>` tags. Substitution and match rules are consulted in the order in which they are listed within the `<regex_map>` element. Each rule may assign values to variables.

Rule Syntax

Every `<subst>` or `<match>` rule expresses the following logical operation:

If all the regular expressions within this rule match, then perform all of this rule's variable assignments; otherwise, ignore this rule and consult the next rule.

Execution terminates when the first `<match>` rule has been applied, or when there are no more rules. A `<subst>` rule that has been satisfied does not terminate execution (unless it is the last rule).

All attributes of rules use the form `reg_varname` or `val_varname`.

- `reg_varname` attribute—Applies a regular expression to `varname`.
- `val_varname` attribute—Assigns a value to `varname` if all of the regular expressions in the current rule are satisfied.

The following are some simple examples of rules.

- If `vpath` starts with `"/default/main/"` set the encoding to `"8859_1"` and continue processing:

```
<subst reg_vpath="^/default/main/" val_encoding="8859_1"/>
```
- The encoding of all files named `"index_zh_TW.html"` anywhere in the `/web` branch is `"Big5"`. There are no exceptions to this rule, so stop processing if it applies.

```
<match reg_vpath= "^/web/(STAGING|WORKAREA|EDITION).*/index_zh_TW.html"
      val_encoding= "Big5"/>
```

Note that the “or” capability of regular expressions, expressed by the pipe character (`|`), enables this single rule handle three cases at once (`STAGING` or `WORKAREA` or `EDITION`).

- The encoding is always "Shift_JIS".
`<match val_encoding="Shift_JIS"/>`

When there are no `reg_` conditions, the assignment always executes if the rule is encountered. Any rules that occur after this statement are unused.

Regular Expression Syntax

The `regex_map` interpreter uses Perl-Compatible Regular Expressions (PCRE) as its regular expression engine. The PCRE is similar to the Perl regular expression engine and includes advanced features such as lookahead assertions.

Regular expressions in `regex_map` are case-sensitive by default. If a variable is listed in the `opt_case_insensitive` attribute of `<regex_map>`, all regular expressions applied to that variable in the `regex_map` are case-insensitive.

For example, because filenames and URLs are case-insensitive on Microsoft Windows, the following declaration would be recommended when writing a `regex_map` for a TeamSite server on Microsoft Windows:

```
<regex_map opt_case_insensitive="vpath url">
  <subst reg_vpath="..." .../>
  <match reg_url="..." .../>
</regex_map>
```

Variables

Variables store strings to be passed in the following ways:

- as input to a `regex_map` from an application
- from rule-to-rule within a `regex_map`
- as results from a `regex_map` to the application

Variable names are case-sensitive and must begin with a letter and may contain any sequence of alphanumeric characters and the underscore character ("_"). References to any variable whose value is not set by the application or by rules in the `regex_map` evaluates to an empty string.

Application Variables

Any application that uses a `regex_map` gives it a set of inputs before execution and inspects a set of output variables when the `regex_map` processing completes. These input and output variables are known as **application variables**.

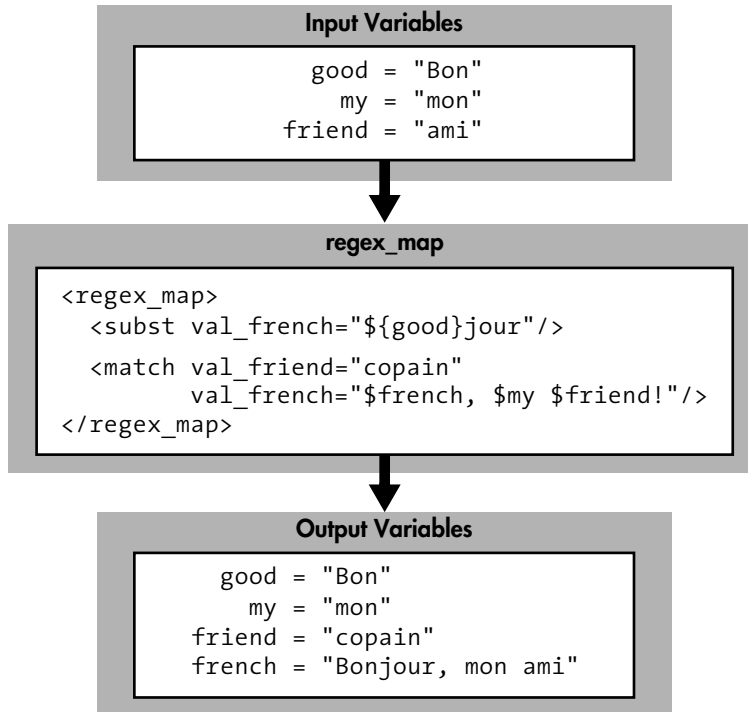
Intermediate Variables

You may find it helpful to assign intermediate results to your variables in `regex_map` rules before arriving at final output values. These **intermediate variables** can help make a complex set of rules more manageable by factoring out several separate conditions into one condensed case. You can then write one rule to act on the condensed case, instead of repetitively writing the same actions for the individual initial conditions. “Strategies for Effective `regex_maps`” on page 280 contains an example of factoring.

Intermediate variables should have names that begin with `x_` to avoid conflicts with application variables that Interwoven may create in the future.

Interpolation of Variables and Captured Subexpressions

When assigning a value to a variable, the values of other variables can be included. In `val_` attributes, each occurrence of `${varname}` or `$varname` causes the value of `varname` to be inserted instead, as shown in the following example:



In the preceding example:

- In the `<subst>` rule, curly braces (`{ }`) are required to separate the variable name `good` from the literal string `jour` that immediately follows.
- In the `<match>` rule, there is no need to disambiguate the three variables because the variable names `french`, `my`, and `friend` are followed by a comma, a space, and an exclamation point, none of which can be confused as being part of a variable name.
- In the second rule, the values of `friend` and `french` are taken from the time at which the rule was encountered. All assignments in a rule appear to occur simultaneously and do not affect each other.



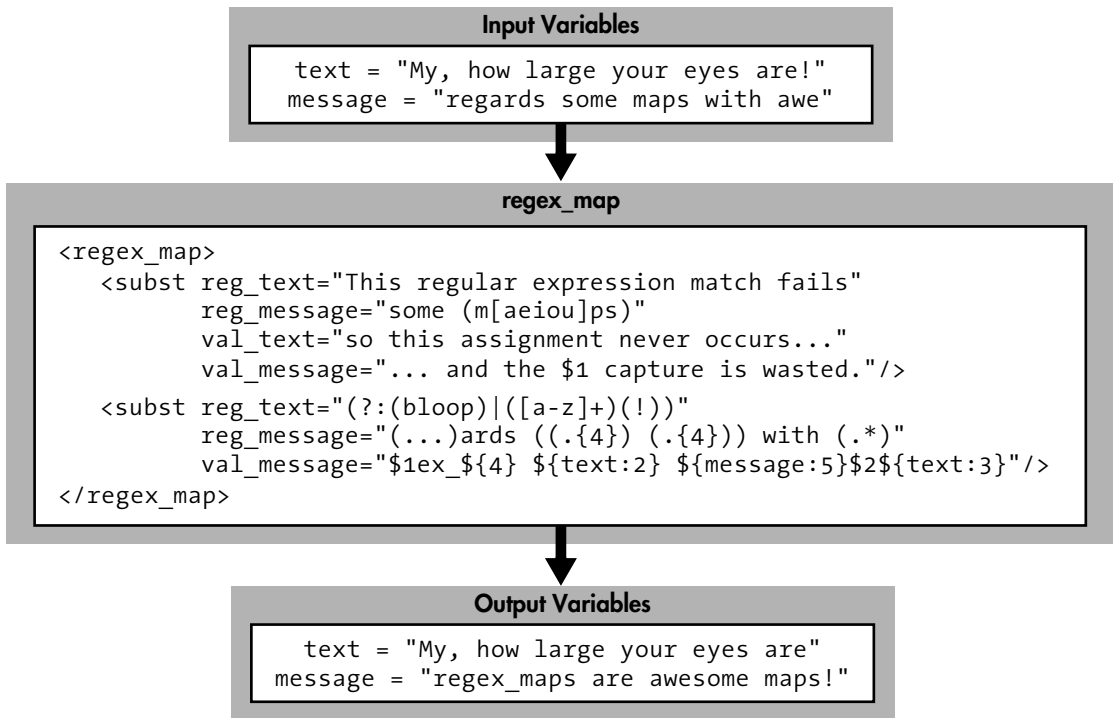
It is also possible to include just portions of variables. Placing parentheses around portions of a regular expression applied to `varname` creates a **captured subexpression** variable that can be used when assigning values. The longhand form of a captured subexpression variable is `${varname:n}`, which refers to the string captured by the n^{th} pair of parentheses in `reg_varname`.

Note: Pairs of parentheses are numbered according to the order in which their left parenthesis occurs within the regular expression. Parentheses of the form `(?:some_expression)` are used solely for grouping characters in the regular expression, not for capturing text during matching, and are excluded from the count.

The shorthand version of a captured subexpression variables is `$n`. Note that the shorthand notation can only be used when the variable being modified is the same as the variable from which the subexpression was captured.

Unlike application and intermediate variables, captured subexpression variables are scoped to the `<subst>` or `<match>` rule that created them. If a captured subexpression variable needs to be used in a subsequent rule, it should be stored in an intermediate variable.

For example, the pair of rules in the following `regex_map` makes the assignment `message="regex_maps are awesome maps!"` in an inefficient way:



In the previous example:

- The first rule does not apply because the value of the text variable does not match the regular expression in `reg_text`.
- While performing the regular expression match for message, the special variable `${message:1}` (the \$1 variable associated with message) takes on the value `maps` within the scope of the rule. However, since the entire rule is inapplicable, it has no effect. Neither of the two `val_assignments` happens, and the temporary `${message:1}="maps"` binding is discarded.



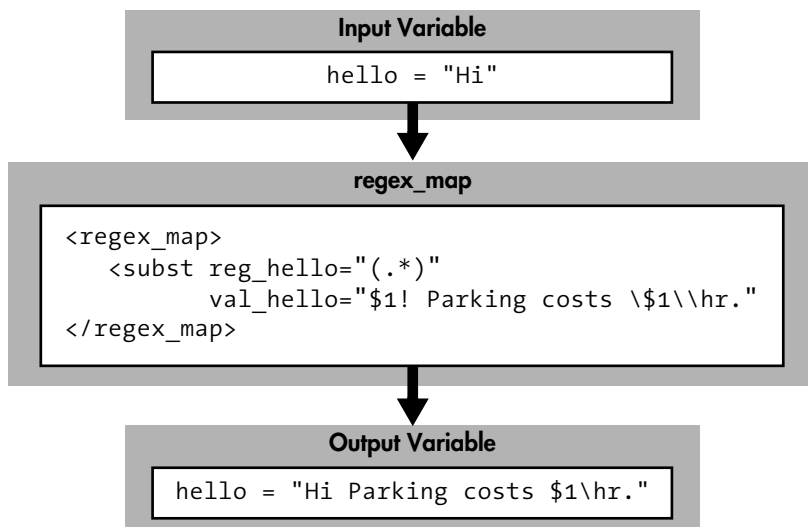
- The second rule does apply, since both of the `reg_text` and `reg_message` matches succeed. The parentheses also capture the text in the strings, resulting in the following temporary bindings:

```
#{text:1} = empty string
#{text:2} = are
#{text:3} = !
#{message:1} = reg
#{message:2} = some maps
#{message:3} = some
#{message:4} = maps
#{message:5} = awe
```

- Finally, variable interpolation occurs for the `val_message` assignment. Since the `$1`, `#{4}`, and `$2` occur in a `val_message` context, they are treated as shorthand for `#{message:1}`, `#{message:4}`, and `#{message:2}`, respectively. The curly braces for `#{4}` are optional in this case, and could be used in other situations to clarify where the variable name ends and literal text begins.

Quoting

Inside a `val_` attribute, dollar signs (\$) have special meaning—they mark the start of captured subexpression names. To force a dollar sign to lose this special meaning (and be treated as a literal dollar sign), it must be escaped by preceding it with a backslash. Similarly, a backslash is treated as a special quoting character unless it is escaped by a preceding backslash.



In the preseding example, `hello` is assigned the value `"Hi! Parking costs $1\hr."` (with the deliberately “wrong” backslash used instead of a forward slash for demonstration purposes).

Also, because `regex_map` is written in XML, characters with special meaning in XML need to be represented using XML entities. These special characters are described in the following table.

Special XML Character	Visual Representation	XML Entry
Double quote	"	"
Apostrophe	'	'
Ampersand	&	&
Greater than	>	>
Less than	<	<

For example,

```
<subst val_statement="Programmers think &quot;1 &amp; 1 is 1.&quot;"/>
```

assigns the following value to the `statement` variable:

```
Programmers think "1 & 1 is 1."
```



Strategies for Effective regex_maps

The regex_map grammar is a powerful string manipulation language yet still allows simple configurations to be expressed simply. This is due to:

- the ability to work with multiple variables
- the use of regular expressions with the capability to reference captured subexpressions
- the option to chain rules with <subst> or stop processing with <match>

By taking advantage of these features, you can write configuration files that are compact and manageable.

The following example demonstrates how factoring and intermediate variables can make a regex_map configuration scale to handle complex situations. Suppose that a system-wide reorganization forced you to rename all files named README to README.TXT and relocate all files under the /a/b branch to the /c/d branch. You could list all of the possibilities as follows:

```
<regex_map>
  <!-- Handle both branch move and file extension addition ->
  <match reg_vpath= "/a/b/((WORKAREA|EDITION|STAGING).*)README$"
    val_vpath= "/c/d/$1README.TXT"/>

  <!-- Handle branch move only ->
  <match reg_vpath= "/a/b/((WORKAREA|EDITION|STAGING).*)"
    val_vpath= "/c/d/$1"/>

  <!-- Handle file extension addition only ->
  <match reg_vpath= "(.*)/README$"
    val_vpath= "$1/README.TXT"/>
</regex_map>
```


But this strategy could become extremely complicated if there were more combinations. A factored set of rules can handle each change independently:

```
<regex_map>
  <!-- Handle a possible branch move -->
  <subst reg_vpath= "/a/b/((WORKAREA|EDITION|STAGING).*)"
        val_vpath= "/c/d/$1"/>

  <!-- Handle a possible file extension addition -->
  <subst reg_vpath= "(.*)/README$"
        val_vpath= "$1/README.TXT"/>
</regex_map>
```

A complicated set of rules could be clarified with intermediate variables, for example:

```
<regex_map>
  <!-- Decompose vpath into branch, area, directory, filename -->
  <!-- Decomposition could be done in just one rule, -->
  <!-- but we choose to break it up with the help of x_rest. -->
  <subst reg_vpath="^(.*?)/((?:WORKAREA|EDITION|STAGING).*)"
        val_x_branch="${vpath:1}"
        val_x_rest="${vpath:2}"/>
  <subst reg_x_rest="((?:WORKAREA|EDITION)/[^/]+|STAGING)(.*)"
        val_x_area="${x_rest:1}"
        val_x_rest="${x_rest:2}"/>
  <subst reg_x_rest="(.*)/(.*)"
        val_x_dir="${x_rest:1}"
        val_x_file="${x_rest:2}"/>
  <!-- End decomposition -->

  <!-- Do the transformations -->
  <subst reg_x_branch="^/a/b$"
        val_x_branch="/c/d"/>

  <subst reg_x_file="~/README$"
        val_x_file="/README.TXT"/>
  <!-- End transformations -->

  <!-- Put vpath back together. -->
  <subst val_vpath="$x_branch$x_area$x_dir$x_file"/>
</regex_map>
```

In the preceding example, factoring out the `vpath` decomposition logic simplifies the actual transformation rules. In a complex situation with many transformation rules, adding a few standardized rules at the beginning and end of the `regex_map` is worthwhile.

“Advanced `regex_map` Example” on page 285 demonstrates of the expressiveness of `regex_maps` by showing how a Roman numeral can be incremented with sequential string substitution rules.

Internationalization and `regex_map`

TeamSite `regex_maps` should be written in UTF-8 and should provide UTF-8 input values and expect UTF-8 output values for all variables. The regular expression engine is UTF-8-aware. For example, a period (.) in a regular expression matches a single character, regardless of the number of bytes needed to represent that character.

Note: If you need to specify non-ASCII literal characters in your `regex_maps`, ensure the text editor you are using can edit and save the `file_encoding.cfg` in UTF-8 encoding. Refer to page 313 for details about text editor encodings.

SmartContext Editing and `file_encoding.cfg`

To determine the encoding of a text file, SmartContext Editing mimics the behavior of your web browser by performing the following series of checks:

- First, SCE checks the Content-Type header from the content webserver. If the MIME type (`text/html` or `text/plain`) is followed by a character encoding declaration (for example, `Content-Type: text/plain; charset=UTF-8`), it uses the specified encoding.
- If the file is an HTML document, SCE searches for a character encoding declaration in an HTML META tag (for example, `<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Big5">`), which it uses if found.
- If the aforementioned methods do not return the encoding, SCE computes the encoding using the `regex_map` configuration in the `file_encoding.cfg` file. It uses the `vpath` as the input variable and the encoding as the expected output.

Source Differencing and Merging and file_encoding.cfg

Unlike SCE, the Source Differencing and Interwoven Merge tools do not mimic your web browser. Instead, they rely entirely on the `file_encoding.cfg` file to determine the character encoding of text documents. The Source Differencing and Interwoven Merge tools assume that the “other” file and the “common ancestor” file share the character encoding of the workarea file.

The following list of encodings are the IANA preferred charset names (<http://www.iana.org/assignments/character-sets>) and are valid entries for the `file_encoding.cfg` file:

English, French, German:

- ISO-8859-1
- ISO-8859-15
- windows-1252

Japanese:

- Shift_JIS
- EUC-JP

Unicode:

- UTF-8

Note: `file_encoding.cfg` has no effect on the file encoding seen in visual differencing. This is so that what is seen in visual differencing tool most closely approximates what will be seen in the production environment.



Sample file_encoding.cfg

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Input variable: vpath -->
<!-- Output variable: encoding -->
<vpath_to_encoding_map>

  <!-- Ignore upper and lower case when
        evaluating reg_vpath and reg_encoding conditions. -->
  <regex_map opt_case_insensitive="vpath encoding">
    <!-- Set the default result. A default like this is highly recommended. -->
    <subst val_encoding="8859_1"/>

    <!-- Make a note of Japanese files scattered about. -->
    <subst reg_vpath="(?:_ja|_jp|_jpn)\."
          val_x_lang="Asian:Japanese"/>

    <!-- Likewise with Chinese files. -->
    <subst reg_vpath=".*\.zh\.txt$"
          val_x_lang="Asian:Chinese"/>

    <!-- As site policy, our Japanese files are Shift-JIS -->
    <subst reg_x_lang="Asian:Japanese"
          val_encoding="Shift-JIS"/>

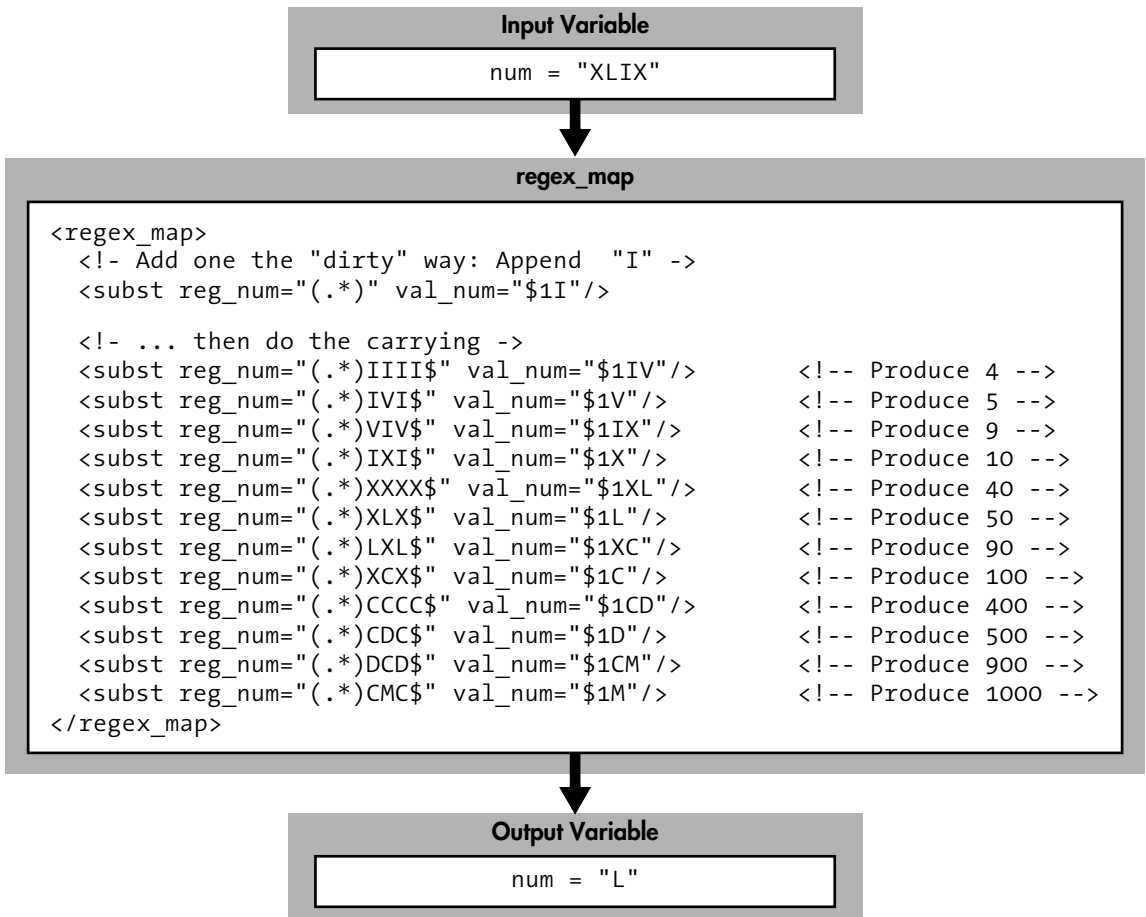
    <!-- As site policy, our Chinese files are Big5 -->
    <subst reg_x_lang="Asian:Chinese"
          val_encoding="Big5"/>

    <!-- Otherwise, the directory name at the top of the area is the result. -->
    <subst reg_vpath="(?:(:WORKAREA|EDITION)/[^/]+|STAGING)/([^\s]+)/"
          val_encoding="${vpath:1}"/>

    <!-- Canonicalize encoding names. Try Shift_JIS, then SJIS. -->
    <match reg_encoding="(sjis|shift[_-]jis)"
          val_encoding="Shift_JIS"/>
  </regex_map>
</vpath_to_encoding_map>
```

Advanced regex_map Example

This example `regex_map`, which adds one to a Roman numeral, demonstrates the power of chained substitution rules. It uses `num` as both the input and the output variable. The example below shows 49 being transformed into 50.



The `regex_map` language works well with Roman numerals because it is designed for string manipulation. It would be much more difficult to write a `regex_map` that increments Arabic numerals, due to the larger set of rules needed to increment a single digit and the lack of looping capability to perform the carrying operation.

Appendix C

High Availability TeamSite

This appendix describes TeamSite HA (High Availability). TeamSite HA has two aspects

- A watchdog daemon that monitors the TeamSite server.
- A “hot standby” integration with Microsoft Clustering.

You must purchase TeamSite HA in addition to the base version of TeamSite to use the features described in this appendix.

HA Watchdog

About HA Watchdog

HA Watchdog uses a watchdog daemon and a set of associated tools and scripts to monitor the TeamSite server, detect process and power failures, log failure events, and optionally take corrective action. After TeamSite HA is installed and configured, it is transparent to end users, performing all user-visible operations in a way that is identical to the base version of TeamSite.

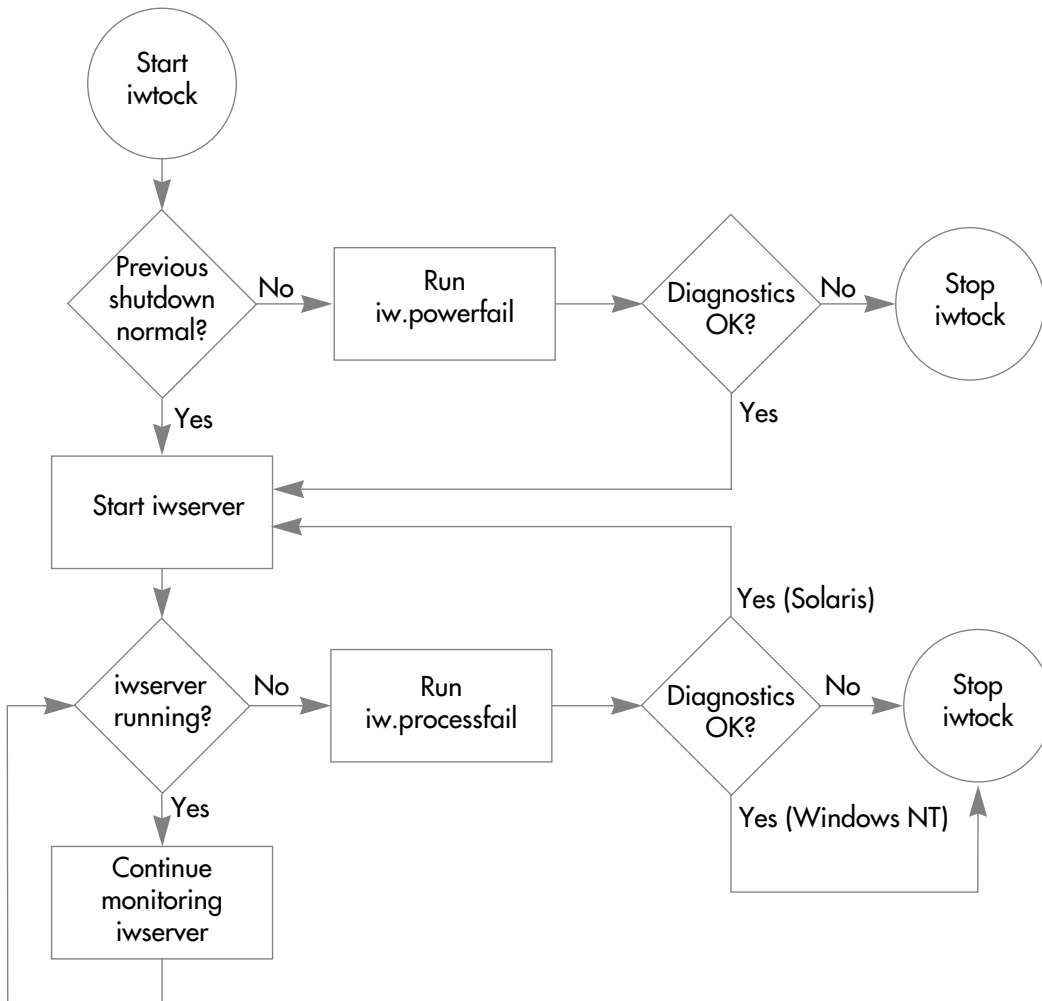
You can configure the HA Watchdog in several ways. For example, you can instruct it to:

- Shut down the TeamSite server and notify a specified system user that a power or process failure was detected.
- Stop the TeamSite server after detecting a failure even if all post-failure system checks are normal.
- Perform a different action depending on whether a power failure or a process failure was detected.
- Perform any other Perl script-defined action automatically upon failure detection.

You can also turn off the availability module completely, in which case the base version of TeamSite continues to run.

TeamSite HA Watchdog Components and Processes

The following flowchart shows the TeamSite HA Watchdog components and processes. See the section immediately following the flowchart for an explanation of each item.



The main TeamSite HA Watchdog component is the `iwtock` watchdog daemon, which starts the `iwserver` process and tracks `iwserver` execution for as long as TeamSite server is running. When `iwtock` first starts, it determines whether the previous TeamSite shutdown was abnormal or normal. If it detects an abnormal shutdown, `iwtock` runs the `iw-home\ha\conf\iw.powerfail` script, which can be configured either to stop `iwtock` or to perform a variety of system checks or other actions as described later in the “Configuring TeamSite HA Watchdog” section. If the system meets the passing criteria defined in `iw.powerfail`, `iwtock` starts `iwserver`. If the system does not meet the passing criteria, `iwtock` stops and `iwserver` is not started. All output from `iw.powerfail` is logged in `iwserver.log`.

If `iwtock` determines that the previous TeamSite shutdown was normal, it starts `iwserver`. From this point on, `iwtock` continues to monitor `iwserver`. If at any time `iwtock` detects that `iwserver` is not running and there is no evidence of an explicit shutdown, it assumes that an unexpected shutdown or system interruption has occurred. In this situation, `iwtock` runs the `iw-home\ha\conf\iw.processfail` script, which can be configured either to stop `iwtock` or to perform a variety of system checks or other actions as described in “Configuring TeamSite HA Watchdog” on page 290. If the system meets the passing criteria defined in `iw.processfail`, `iwtock` exits (it cannot start `iwserver` due to Windows NT architecture characteristics). If the system does not meet one or more passing criteria, `iwtock` stops and `iwserver` is not restarted.

All output from `iwtock`, `iw.powerfail`, and `iw.processfail` is logged in `iwserver.log`.

Notes: If `iwserver` attempts to spawn more than once within 30 seconds of initial startup or a restart, `iwtock` will exit.

On any list of active system processes, `iwtock` appears as `iwperl` (you will not see a list entry called `iwtock`).

Installing TeamSite HA Watchdog

Perform the following steps to install TeamSite HA Watchdog. After the installation is complete, you have access to the failsafe and availability modules.

1. Log in as **Administrator**.
2. In the TeamSite HA distribution media window, click **TeamSiteHA.exe**.
3. After the program executes, reboot the system.

Configuring TeamSite HA Watchdog

Configuring TeamSite HA requires that you edit the `iw.powerfail` and `iw.processfail` scripts to execute tasks that are relevant and specific to your installation. Details are as follows.

iw.powerfail

The default `iw.powerfail` script shown here is shipped with TeamSite. In its current form, it only logs its own name (`iw.powerfail`) when executed. It also contains a commented example of how you could configure the script to run the `iwsi` and `iwfailsafe` programs, and send email to a system administrator when the script executes. You can configure this script to perform any action upon execution; the only requirement is that you use Perl syntax compatible with Perl Release 5.00503. All results returned by `iw.powerfail` are logged in `iwserver.log`.

Note: To force `iwtock` to exit rather than start the TeamSite server after `iw.powerfail` executes, specify an `iw.powerfail` exit value of 127. This feature is included for scenarios in which TeamSite should not restart automatically following a power failure.

```

use File::Basename;
print( basename( $0 ) . "\n" );

#
# Use this script to execute processes that can clean up after a powerfail
# crash.
#
# This script is executed when the Watchdog daemon determines that the
# system was not taken down cleanly, and the daemon is itself beginning
# execution.
# Some of the things that might be tried:
#

# iws
#
# iwfailsafe -n # check for backing cache consistency
# iwfailsafe    # if backing cache consistency check passed
#                we should do the recovery and continue with bring up
#

#
# You may also want to mail your system administrator at this point:
#

#use Mail::Send;
#msg = new Mail::Send Subject=>'TeamSite problem', To=>'admin,root';
#mfh = $msg->open;
#print $mfh "Please address TeamSite issues at your earliest convenience";
#$mfh->close;

#
# If, after executing the backing store utilities, you do not wish to
# continue bringing up the system, then exit this script with a 127.
# 127 indicates to the daemon that it is not to continue with the bringup.
#

#exit 127

```

**iw.processfail**

The default `iw.processfail` script shown here is shipped with TeamSite. In its current form, it only logs its own name (`iw.processfail`) when executed. It also contains a commented example of how you could configure the script to run the `iwsi` and `iwfailsafe` programs, and send email to a system administrator when the script executes. You can configure this script to perform any action upon execution; the only requirement is that you use Perl syntax compatible with Perl Release 5.00503. All results returned by `iw.processfail` are logged in `iwserver.log`.

Note: Because of the way in which TeamSite and Windows NT interact, it is not possible for `iw.processfail` to restart the TeamSite server. Instead, the `iwtock` daemon exits whenever `iw.processfail` finishes executing. At that point, you must reboot the system to restart the TeamSite server.

```

use File::Basename;
print( basename( $0 ) . "\n" );
#
# Use this script to execute processes that can clean up after a TeamSite
# crash after the system has begun processing data.
#
# This script is executed when the Watchdog daemon determines that the
# system was not taken down cleanly, and the daemon has already begun
# observing the execution of TeamSite.  Some of the things that might be
# tried:
#
#
# iws
#
# iwfailsafe -n # check for backing cache consistency
# iwfailsafe    # if backing cache consistency check passed
#                we should do the recovery and continue with bring up
#
#
# You may also want to mail your system administrator at this point:
#
#use Mail::Send;
$msg = new Mail::Send Subject=>'TeamSite problem', To=>'admin,root';
$mfh = $msg->open;
#print $mfh "Please address TeamSite issues at your earliest convenience";
$mfh->close;
#
# If, after executing the backing store utilities, you do not wish to
# restart the system, then exit this script with a 127. 127 indicates
# to the daemon that it is not to restart the server. The server will
# not restart under NT at all.
#
#exit 127

```

Starting and Stopping the Server Under HA Watchdog

You can manually start and stop `iwserver` under TeamSite HA Watchdog just as you would under the base version of TeamSite. See Chapter 7, “Managing the TeamSite Server” for more information. On any list of active system processes, `iwtock` appears as `iwperl` (you will not see a list entry called `iwtock`).

Uninstalling TeamSite HA Watchdog

The following sections describe how to uninstall TeamSite HA and revert to the base version of TeamSite.

1. Log in as **Administrator**.
2. Select **Start > Settings > Control Panel**.
3. Select the **Add/Remove Programs** control panel.
4. Select the **TeamSiteHA** and click **Add/Remove**.
5. When the uninstall completes, reboot the system.

Related Documentation

See the `iwsi` and `iwteamsite` man pages in *TeamSite Command-Line Tools* for more information about TeamSite HA Watchdog.

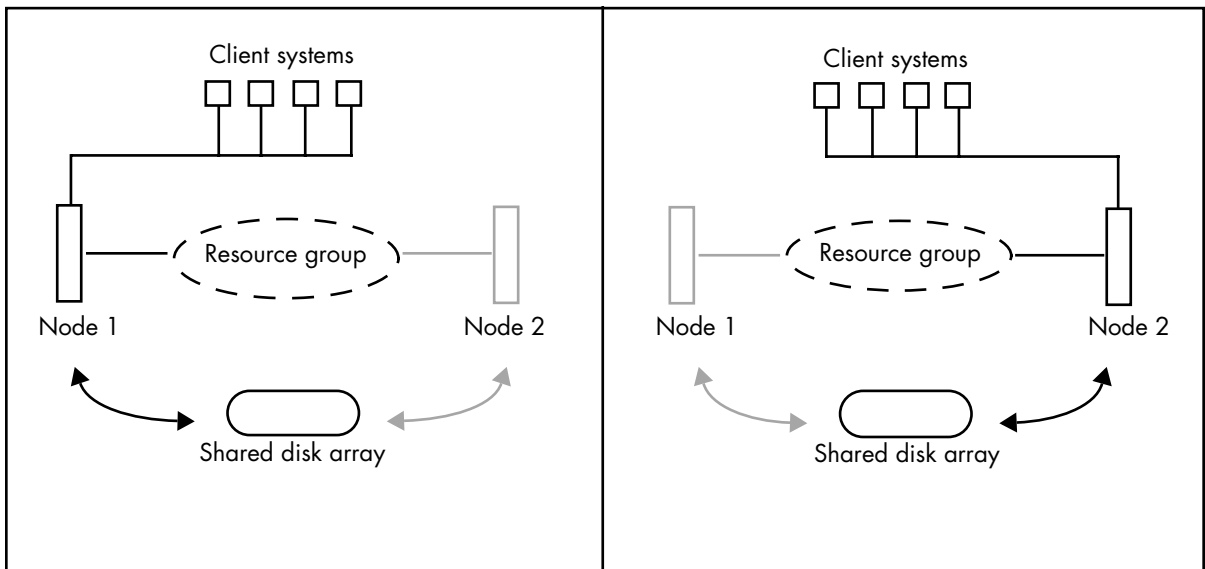
HA Hot Standby

About HA Hot Standby

High Availability Hot Standby integrates TeamSite with Microsoft Cluster Server to provide uninterrupted access to the TeamSite server. For complete information on Microsoft Cluster Server, refer to Microsoft’s *Cluster Administration* manual. For definitions of some of the terms used in the following paragraph, see Definitions in the “About Microsoft Cluster Server” section of this document.

High Availability Hot Standby uses a two system (or *node*), cluster on Microsoft Windows NT and Windows 2000. In this environment, a single *resource group* is created that contains resources needed for TeamSite operation (IP address, location of backing store and backing store copy, network name, TeamSite services). While each node can own and manage cluster resources and a shared disk array containing the TeamSite backing store and a copy of the backing store, only one node can access these at a time. In other words, a node is always active while the other waits passively for a failover to occur.

When any TeamSite service on the active node fails, the passive one becomes active, taking ownership of the cluster's resources and using the last known good copy of the backing store as its backing store. Client requests are then redirected to the currently active node until the failed system is brought back online.



Normal Operation: Node 1, the primary node, owns the resource group containing such resources as a shared IP address, location of the backing store and backing store copy, network name, and TeamSite. The backing store and backing store copy are located on a shared disk array connected to both nodes via SCSI.

Failover: A failure occurs on the primary node. Node 2 takes ownership of the resource group and uses the copy of the backing store as its backing store. Client requests are routed to the second node until the failed node is brought back online.

About Microsoft Cluster Server

Definitions

node

Each node has its own memory, system disk, operating system, and subset of the cluster's resources.

resource

To the cluster service, a resource is any physical or logical component that can be taken offline or brought online, managed in a cluster, hosted by one node at a time, and moved between nodes. Examples of resources are disks, network names, IP address, databases, websites, and application programs.

resource group

A group is a collection of cluster resources. A group is always owned by exactly one node at a time. Likewise, a resource is owned by a single group. These relationships ensure that all the group's member resources reside on the same node. The basic unit of failover for the Microsoft Cluster Server is a group. When one resource in a group fails and it is necessary to move the resource to an alternate node, all the resources in the group are moved to the alternate node.

Architecture

The Microsoft Cluster Server is a multi-node cluster based on the shared-nothing clustering model. Under the shared-nothing model, any node in the cluster can access a resource, but only one node may own and manage a resource at a time. When the active node fails, another node in the cluster takes ownership of cluster resources and assumes the workload of the failed system.

Installing TeamSite and High Availability Hot Standby

Before You Begin

Hardware Requirements

High Availability Hot Standby supports on any hardware configuration listed in Microsoft's Cluster Server Compatibility list. Go to the Microsoft website at <http://www.microsoft.com/hcl/default.asp> and select Cluster from the drop-down menu.

- Only a two node cluster is supported.
- Both nodes must use an Intel CPU (see the *TeamSite Administration Guide* for recommended specifications).
- Shared disk array.

Software Requirements

Verify basic cluster setup, cluster interconnect and failover capabilities before installing TeamSite.

- Both nodes must use Microsoft Windows NT Server Enterprise Edition Version 4.0 or Windows 2000 with Service Pack 5.
- Microsoft Cluster Server version 1.0 must be running on both nodes of the cluster.
- TeamSite should be installed under a similar directory structure on both nodes. For example, if you install TeamSite on Node 1 under C:\Program Files\Interwoven, you must install TeamSite on Node 2 also under C:\Program Files\Interwoven. Therefore, make sure that at least one local drive on each node has the same drive letter as the other node. If this is not possible, try the workaround in “Troubleshooting HA Hot Standby” on page 303.

Preparing to Install High Availability Hot Standby

Prepare the Hot Standby environment:

1. Interwoven TeamSite depends on three cluster resources, one each of the following resource types:
 - IP Address
 - Network Name
 - Physical Disk

Create a resource for each of the above three types, and add them to a Cluster Group. Refer to the Microsoft Cluster Administrator manual(s) for information about resources and how to create them.
2. The TeamSite backing store must reside on a Physical Disk resource that represents a volume in the shared disk array. Refer to your hardware manuals for information on how to create and manage volumes on a shared disk.
3. To move TeamSite from a non-cluster environment to a cluster environment, copy the existing backing store to the volume that represents the physical disk resource.

4. Prepare to reboot the cluster nodes after installing TeamSite.
5. Stop the Cluster Server on both nodes: in the Services control panel, highlight **Cluster Server** and click **Stop**.
6. Change the startup mode for the Cluster Server on both nodes to Manual from Automatic: in the Services control panel, highlight **Cluster Server**. Click **Startup** and select **Manual**.

Installing High Availability Hot Standby

Step1: Install High Availability Hot Standby on the Primary Node

1. In **Control Panel > Services**, verify that the Cluster Server is not running.
2. Start the TeamSite installer as described in Chapter 2, “Installing TeamSite.” You must install TeamSite on a local drive (not on a shared disk).
3. When TeamSite Setup asks you where you want to locate the backing store, enter a directory on a local drive, for example `C:\iw-store`.
4. Complete the installation as described in Chapter 2, “Installing TeamSite.”
5. Reboot and log in to the system. Wait for TeamSite IIS configuration to complete.
TeamSite now runs with a local drive as the backing store.
6. From the Services control panel, stop the Interwoven services (Interwoven TeamSite, Interwoven Proxy, Interwoven GRC, and Interwoven Reports). Change their startup mode from Automatic to Manual.
7. Change the startup mode of Cluster Server service from Manual to Automatic.
8. Start the Registry Editor (`regedt32.exe`).
9. Go to `HKEY_LOCAL_MACHINE\SOFTWARE\INTERWOVEN\TeamSite`. Double click the `iw-store` value in the right pane. Enter the location of the backing store on the shared disk.
10. Reboot.

Step 2: Install High Availability Hot Standby on Node 2:

Perform all procedures in “Step1: Install High Availability Hot Standby on the Primary Node” on node 2.

Step 3: Create Custom Resource Types on the Primary Node

1. On the primary node, select **Control Panel > Services** and verify that the Cluster Server Service on that node is set to Automatic and is currently running.
2. If Cluster Administrator is running on either node, close it.
3. Open a command prompt. Go to the *iw-home\bin* directory.
4. If applicable, perform the workaround described in “Troubleshooting HA Hot Standby” on page 303. Use this workaround only if your system configuration prevents you from having at least one local drive on each node that has the same drive letter as the other node.
5. From the *iw-home\bin* directory, execute the following command to create the Interwoven TeamSite cluster resource type:

```
>iwcrtype -create -type iw-home\bin\IWOVTeamSite.DLL
```

If the install path contains any spaces in it, enclose the path to the DLL in double quotes.

6. Respond Y at the Y/N prompt.
7. Verify that the TeamSite resource type is created successfully.
 - a. Start the Microsoft Cluster Administrator:
Start > Programs > Administrative Tools (Common) > Cluster Administrator.
 - b. Double-click the cluster name.
 - c. Click on **Resource Types**. Make sure that you see Interwoven TeamSite in the right pane with the DLL path pointing to the correct location.
8. In the Cluster Administrator, double-click on the cluster name.
9. Right click on **Resources**. Select **New > Resource**.

10. In the New Resource window, do the following:
 - a. Enter a name for the resource being created (example: TeamSite 4.5). You can optionally enter a description for the resource.
 - b. In the **Resource Type** drop-down menu, select **Interwoven TeamSite**.
 - c. In the **Group** drop-down menu, select the group you created in step one in “Preparing to Install High Availability Hot Standby” on page 297.
11. Click **Next**.
12. Add both nodes to **Possible owners** in the left pane and click **Next**.
13. Add the three resources you created in step one in the “Preparing to Install High Availability Hot Standby” section of this document to **Resource Dependencies**.
14. Click **Finish**.
15. If the resource is created successfully, a message box appears stating “Cluster Resource *resource_name* created successfully.”
16. In the right pane, select the new resource.
17. Right-click the highlighted resource and select **Properties**.
18. Verify each of the following:
 - The name of the resource is correct.
 - Both nodes have been added as “Possible owners.”
 - The resource type is Interwoven TeamSite and the state of the resource is offline.
19. Click the **Dependencies** tab.
20. Verify the following three resource types exist:
 - IP Address
 - Network Name
 - Physical Disk
21. Click the **Advanced** tab and select the **Restart** option.

22. Check the **Affect the group** check box.
23. Enter 0 as the **Threshold** value.
24. Under the **LooksAlive poll interval** section, select **use value from resource type**.
25. Under the **IsAlive poll interval** section, select **use value** from resource type.
26. Click **Apply**.
27. Click **OK**.

Now the resource is ready to be started.

28. Highlight the resource you created, right click, and select **Bring Online**.

TeamSite starts on the node.

Step 4: Verify Secondary Node's Access to Cluster Resources

1. Go to the other node where TeamSite is not currently running.
2. Verify that the Cluster Server Service is configured to start automatically.
3. Verify that all the Interwoven related services are configured to start manually.
4. Reboot the node.
5. After rebooting, start Cluster Administrator on this node and verify that you are able to read the properties of Interwoven TeamSite resource type and properties of the resource you created in "Step 3: Create Custom Resource Types on the Primary Node."
6. Verify that the TeamSite resource status is displayed correctly.

Setting Up Hot Standby Failover

Hot Standby failover refers to the situation in which TeamSite server uses the location that contains the last known good backup copy of the backing store as the current backing store when a failover occurs. This is the case if you do not have High Availability Watchdog implemented on your system. Using widely available hardware solutions, a cluster administrator can set up and prepare a mirror copy of the active backing store as frequently as necessary. This mirrored copy is made available to both nodes in the cluster via a shared disk array. The backing store and the

mirrored copy represent separate volumes on the same disk array. To set up Hot Standby Failover, configure the `iw-store` Registry value on the passive node to point to the location of the mirror copy of the backing store.

To set up Hot Standby, follow these steps:

1. Use the Cluster Administrator to create another Physical Disk resource for the backing store copy that represents a volume in the shared disk array. Add this resource to the same group that contains the TeamSite resource and the physical disk resource that represents the volume for the primary backing store.
2. On the primary node, use the Cluster Administrator to start TeamSite Cluster Resource with the primary backing store.
3. Change `iw-store` under the `HKEY_LOCAL_MACHINE\Software\Interwoven\TeamSite` Registry key on the standby node to point to the location on the shared disk array that contains the backup copy of the backing store.
4. Use a mirroring solution to create a consistent copy of the backing store at pre-determined intervals (frequency of the mirror backup depends on your ability to do a `iwfreeze` or to use a high-speed hardware-level mirroring solution). The backup copy of the backing store should reside on the shared disk array that the Registry entry points to (step 3 above).

When TeamSite fails over to standby node, it will use the copy of the backing store.

After the problem that caused the failover is corrected, you can manually move the TeamSite resource from the secondary node back to the primary node. Before you do this, you will need to sync the primary backing store with the backup backing store if any activity was performed while the standby node was running TeamSite.

Troubleshooting HA Hot Standby

If you are having trouble setting up Hot Standby:

- Verify basic cluster setup, cluster interconnect and failover capabilities prior to installing TeamSite.
- When a failover occurs, or is initiated manually using Cluster Administrator, the node on which TeamSite was originally running must be rebooted before TeamSite can be restarted on that node. You must also reboot the node when you take the TeamSite resource offline and want to bring it online again on the same node. If you do not reboot the node between two starts of the TeamSite resource, the system may hang.
- Never attempt to start TeamSite services using the Control Panel on the passive node. Otherwise, the system may hang.
- Never stop the TeamSite services using the Control panel on the active node. Otherwise, the system may hang.
- The IP Address and Network Name resources must be added as dependencies to the TeamSite resource, and all three resources must be in the same Resource Group. TeamSite clients must connect to the TeamSite server running on either cluster node using the IP address value specified when creating a resource of type IP Address and/or the name specified when creating the resource of Network Name. Clients must re-connect after a failover.
- The system may hang if you attempt to create more than one resource of type Interwoven TeamSite.
- It is standard practice to configure Cluster Server to run under a domain user's account. Refer to the Cluster Administration manual(s) for the configuration appropriate to your situation.
- Whenever the TeamSite resource is brought online, the resource DLL polls for 30 seconds in three 10 second intervals to confirm TeamSite services are running well. When using the Cluster Administrator to bring TeamSite resource online, if you continue to see the resource status as "Online pending," refresh the Cluster Administrator window after approximately a minute.
- Never set Restart Threshold of a resource of type Interwoven Teamsite to a value other than zero. (Restart Threshold can be set using the Cluster Administrator.)
 - a. Highlight the resource on the right pane.
 - b. Right click the highlighted item.
 - c. Select **Properties** from the pop-up menu.
 - d. Click the **Advanced** tab.



The TeamSite resource cannot restart on the same node after a failure unless the node has been rebooted.

- When experiencing problems with TeamSite in the Cluster environment, enable Cluster logging by:
 - a. Setting the environment variable `ClusterLog=path_to_the_log_file`.
 - b. Setting the environment variable `ClusterLogLevel=3` on both nodes and restart Cluster Server.

Setting up HA Hot Standby with Different TeamSite Installation Directories

If the TeamSite installation directory is not the same on both nodes (For example:

C:\Program Files\Interwoven\TeamSite on Node1 and

D:\Program Files\Interwoven\TeamSite on Node 2):

Copy `iw-home\bin\IWOVTeamSite.DLL` to `%SystemRoot%\Cluster` directory. Specify this path to the DLL when creating the Interwoven TeamSite cluster resource type.

Appendix D

Internationalization

TeamSite 5.5.1 is engineered with your global enterprise in mind. This includes internationalizing the TeamSite server to support multibyte languages and locales at the operating system, and localizing the WebDesk user interface and documentation. Internationalized TeamSite supports the following needs:

- International user data—Enables users to enter data, content, and field values in English, Traditional Chinese, Simplified Chinese, French, German, and Japanese.
- Localized operating system—The TeamSite server runs on any one of the following localized operating systems: English, French, German, and Japanese (one locale per instance of `iwserver`).
- Localized user interface—The WebDesk GUI has been localized in French, German and Japanese.
- Localized file names—You are no longer restricted to having file and directory names in ASCII character encoding. File, directory, branch, workarea, and edition names can have Japanese names on Japanese servers, German names on German servers, and French names on French servers.
- Continued support for processing of non-English metadata and Templating content (introduced in TeamSite 4.2.1 and 4.5.1).

Supported Client and Server Platforms

The client connecting to the TeamSite server must use the same language as the server (they can be different locales of the same language). For example, running WebDesk on French Windows 98 connected to a Solaris 2.7 TeamSite server running in the French Latin 1 locale (`fr`) is supported. However, if that same French Windows 98 client logged into a Windows 2000 Japanese TeamSite server, and added files with names containing French

characters, those files would not be supported by the TeamSite server due to limitations with the native operating system and handling of characters outside of its code pages.

Team Site 5.5.1 supports client/server interaction by clients and servers running in the following configurations:

Servers

- Windows 2000 Service Pack 1 or higher (US English, French, German, Japanese)
- Windows NT 4.0 Service Pack 6 (US English, Japanese)

Clients

- Windows 98 (US English, French, German, Japanese)
- Windows NT (US English, French, German, Japanese)
- Windows 2000 (US English, French, German, Japanese)
- Solaris 2.6, 2.7, and 2.8 US (English)
- MacOS 9.x (US English, Japanese)

Browsers

- Internet Explorer 5.x (Internet Explorer 6 is not supported)
- Netscape 4.76 through 5.x (Netscape 6 is not supported, Netscape is not supported on MacOS 9.x)

Refer to the table on page 27 for more information about browser compatibility.

Note: Appendix E, “Client/Server Compatability” contains a series of tables that contain information about client/server/language compatibility for all supported platforms; WebDesk UI, Launchpad, and Templating UI; file, directory, backing store, branch, and workarea names; and Templating content and metadata.

Supported TeamSite Server Locales

The following table describes the supported TeamSite server locales:

Language	Server Locale Supported
Japanese	Japanese NT and Japanese 2000
German	German 2000
French	French 2000
English	U.S. English NT and U.S. English 2000

The TeamSite `iw.cfg` file now contains a `server_locale` entry in the `[iwserver]` section. The entry specifies the locale in which current execution of the TeamSite server (`iwserver`) is running. For detailed information about the `server_locale` setting, refer to “Configuring the TeamSite Server Locale” on page 154.

Supported Content

TeamSite supports non-ASCII characters in branch, area, directory, `vpath`, and file names in addition to the contents of a file.

Localization Overview

The following sections list whether or not major TeamSite features have been translated. These features are described throughout the TeamSite documentation.

What’s Been Translated?

The following TeamSite 5.5.1 features have been translated into French, German, and Japanese:

- TeamSite login screen
- WebDesk user interface and online help
- WebDesk submit workflow
- Visual Format

- LaunchPad applet
- Java Templating client installer
- TeamSite Templating (browser-based and Java Templating) and Visual Format
- SmartContext Editing (SCE)
- The following printed documentation (including PDF versions on TeamSite CD-ROM):
 - *TeamSite Author's Guide*
 - *TeamSite User's Guide*
 - *TeamSite Templating User's Guide*

What's Not Been Translated?

The following TeamSite 5.5.1 features are available in English only:

- TeamSite installer
- TeamSite Templating server installer
- WebDesk Pro user interface and online help (see note below)
- LaunchPad application (no longer supported on any operating system)
- Metadata capture form
- Command-line Tool user interface
- The remainder of the printed documentation and Release Notes
- WorkflowBuilder application (WorkflowBuilder only runs on US clients and servers)

WebDesk Pro GUI elements, including buttons and drop-down menus, retain English names but may look slightly different because all HTML pages of our browser-based GUI are UTF-8 encoded, even for US English installations. Your client browsers may therefore choose different fonts to render UTF-8 encoded HTML pages.

TeamSite users can interact with the GUI using any one of the supported languages, but the TeamSite server must be listed in “Supported TeamSite Server Locales” on page 307.

Limitations and Assumptions

- An internationalized TeamSite server does not mean that your TeamSite server can be run in multiple locales concurrently. The TeamSite server can run in any supported locale, but one locale at a time.
- It is expected that the locale in which the TeamSite server runs is the same locale as the rest of file system and server operating systems. Consider the following scenario:
 - a. You have a file server which runs in `ja` (Japanese Extended UNIX Code) locale, with a hierarchy of file and directory structures with names encoded in Japanese EUC.
 - b. You install and run your TeamSite server on this file server.
 - c. You use the file system interface to migrate your existing hierarchy of files and directories into TeamSite's Intelligent File System (`/iwmnt`).
 - d. The TeamSite server must run in a `ja` locale for these file and directory names to be processed correctly. If you change the locale to `ja_JP.PCK` (Japanese Shift-JIS) before TeamSite server is started, the TeamSite server would interpret the imported file and directory names as `ja_JP.PCK` encoded. This is not a supported scenario.
- Mixed-locale file systems are not supported. For example, a scenario where a parent directory has directory names encoded in `ja_JP.PCK` (Japanese Shift-JIS), and child directories have file names encoded in `ja` is not supported.
- If TeamSite server is running on a German operating system using a German `Latin1` locale, it is possible to create a branch or workarea on the TeamSite Intelligent File System with Japanese names using the TeamSite GUIs. However, when viewed with the file system interface, these Japanese names would appear as illegible characters because the server is running in a `Latin1` locale and does not include the Japanese character set. This is not a supported scenario.

Note that this scenario is supported for Metadata because Metadata entered using the TeamSite GUIs does not interact with server operating system. Any data that is interchanged with the server operating system (including `VPATHs`) are only meaningful if they are within the server locale's encoding.

- If TeamSite Intelligent File System is functioning as a networked file server, it is expected that all other networked file system clients (for example, NFS clients) are operating in the same locale as the TeamSite Intelligent File System file server.

Currently, NFS does not enforce this restriction and therefore enables NFS clients to be in a different locale than the NFS server. However, NFS protocol does not do encoding conversion. Therefore, file and directory attributes (including names) are passed through in binary format. This would not work for TeamSite IFS functioning as file server because it does encoding conversion from and into UTF-8 based on the server file system's locale.

Backing Stores and Character Encoding

User-defined backing stores which are named using multibyte characters, must have a corresponding entry in the `iw.cfg` file. Refer to “Creating Multiple Backing Stores” on page 251 for detailed information.

About UTF-8

UTF-8 is the 8-bit encoding format for Unicode. Unicode is a system for exchanging, processing, and displaying diverse written languages. Unicode supports the principal written languages of the world as well as many classical languages.

Interfacing with Localized Operating Systems

The internationalized TeamSite server's virtualized Intelligent File System (IFS) functions the same way a regular file system does on localized operating systems. For example, if TeamSite runs on a server that is running in the EUC-JP locale, the TeamSite IFS is displayed and functions as an EUC-JP encoded file system.

To achieve this, TeamSite system calls to the operating system are converted from UTF-8 encoded textual data (for example, VPATH information) into the locale of `iwserver` (as defined by the `server_locale` setting in `iw.cfg`). In most cases, this is the same as the operating system's native locale. The conversion is also required when operating system information is returned to TeamSite.

Note: If the TeamSite server is run in a different locale than the host operating system's locale, the TeamSite virtual file system would use a different encoding locale compared to the rest of host server's file systems. By default, the TeamSite server locale uses the native locale of the host operating system.

Accessing the Localized Interface

To display the localized (French, German, or Japanese) WebDesk interface, you must change your browser's language settings to the appropriate language.

To display the localized (French, German, or Japanese) LaunchPad and Java Templating interfaces, your client operating system must be in the same locale as the interface you want to display.

CLT Internationalization

Command-line tools are now locale-sensitive such that arguments passed into the CLT as textual arguments—including submit comments and VPATH specifications—can be text characters from any of the supported languages (English, French, German, or Japanese). For example, if you typed submit comments into the `iwsubmit` CLT in Japanese, the character encoding of the submit comments would depend on the locale under which `iwsubmit` was executed.

When CLTs are executed, the locale is determined by referencing the System Locale setting in the Control Panel's Regional Options. Based on this locale, it determines how to interpret character encoding of the textual arguments passed from the command-line.

By default, the TeamSite Server (like most Windows applications) uses code page 1252 on Single Byte Character Set (SBCS) systems in English, French, and German. However, the DOS command window uses different OEM code pages for SBCS English systems and German and French systems. This difference causes TeamSite CLT output to be displayed incorrectly.

Single Byte Character Set System Language	Default Code Page	DOS Command Window OEM Code Page
English	1252	437
German and French	1252	850



To avoid this issue, complete the following procedure before running any TeamSite CLT on SBCS systems.

1. Open the DOS command window.
2. Right-click on the command window's title bar and choose **Properties** from the menu.
The Properties dialog box is displayed.

3. Click the **Font** tab.

4. Select **Lucida Console** font from the **Font** list, and click **OK**.

The Apply Properties dialog box is displayed.

5. Select **Save properties for future windows with same title**, and click **OK**.

6. At the DOS prompt, type **chcp** and press Enter.

The system returns the active code page number: 437 for English systems, or 850 for German and French systems. Record the code page number so you can revert to the default code page for commands that require it.

7. Type **chcp 1252** and press Enter to change the code page to 1252.

The system confirms the active code page is set to 1252. All command window input and output will use this code page.

CGI Internationalization

Since the 4.2.1 release, TeamSite CGIs have used UTF-8 encoding to serve pages to browsers. This causes browsers to return data to TeamSite encoded in UTF-8. This enables TeamSite to support metadata in any language or native encoding. TeamSite 5.5.1 uses this same methodology with VPATHs in addition to metadata.

Specifying File Encoding of Text Files

All browsers rely on default settings to “guess” the encoding of web pages whose encoding is not explicitly declared. If the browser’s default setting is different than that of the actual encoding of the page passed to the browser, the browser may render the page incorrectly. Therefore, the best practice is for your web pages to always declare their encoding. This prevents your browser from guessing incorrectly when you use TeamSite, and ensures that your web site viewers’ browsers will not have to guess which encoding they should use.

For HTML documents, Smart Context Editing (SCE) honors the encoding specified by the `charset` parameter in either a `Content-Type` HTTP header or in an HTML META tag. For example:

- `Content-Type: text/plain; charset=UTF-8`
- `<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=UTF-8">`

To display multibyte characters in non-HTML text documents in SCE with the desired character encoding, the content webserver must be configured to return a `Content-Type` HTTP header that specifies the encoding, for example:

```
Content-Type: text/plain; charset=UTF-8
```

If the `charset` is not specified—either by the content webserver’s `Content-type` HTTP header, or by the `charset` tag within the file—SCE assumes that the document is encoded in ISO-8859-1, which may cause the document to be displayed with “garbage” characters.

Note: To solve the issue of text files that do not specify their encoding, TeamSite 5.5.1 has introduced a new configuration file called `file_encoding.cfg`. Please refer to Appendix B, “Specifying Content Encoding” for detailed information about creating configuring settings in `file_encoding.cfg`.

Text Editor Encodings

The following table shows the default settings for various text editors and how to modify them to use UTF-8 encoding.

Text Editor	Platform	Default Encoding	To Save as UTF-8 Encoding:
Notepad	Windows 2000	ANSI (relative to the localized operating system)	<ul style="list-style-type: none"> Select File > Save As. Select UTF-8 in the Encoding drop down menu.
Wordpad	Windows 2000	Rich Text Format (RTF) (relative to the localized operating system)	<ul style="list-style-type: none"> Select File > Save As. Select Unicode Text Document in the Save as type drop down menu.
	Localized versions of Windows NT 4.0	ANSI (relative to the localized operating system)	Cannot save or render text as UTF-8.

Behavior of Netscape Navigator

If a Netscape browser finds a UTF-8 page, it uses UTF-8 as its default encoding for pages that do not specify their encoding. This may cause the browser to display pages incorrectly if the user browses pages that do not specify their encoding, or creates pages without specifying the encoding.

Scenario 1

1. A Japanese user goes to a Japanese site which does not specify its encoding. Netscape defaults to Japanese (Auto-Detect).
2. The Japanese user logs into TeamSite (UTF-8 pages). Netscape switches to UTF-8.
3. The Japanese user opens a new window and returns to the Japanese site which does not specify its encoding. Now Netscape defaults to UTF-8.

This would not happen if the site specified the encoding of its web pages.

Scenario 2

1. A Japanese user logs into TeamSite (UTF-8 pages). Netscape switches to UTF-8.
2. The Japanese user's content in TeamSite does not include the 'Content-type' META tag.
3. Upon entering SmartContext QA, Netscape tries to render the content as UTF-8, which is probably wrong. The solution to this problem is to always specify the encoding for all HTML content.

Configuring Netscape for Multibyte Characters

Complete the following procedure if you are using a Netscape browser to display multibyte characters:

1. Open your Netscape browser.
2. Select **Edit > Preferences** to display the Preferences dialog box.
3. Click **Appearance > Fonts** to display the Fonts settings.
4. Set the **For the Encoding** field to Unicode.
5. Set the **Variable Width Font** field to a font which supports the language you want to use.
6. Set the **Fixed Width Font** field to a font which supports the language you want to use.
7. Click the **Use my default fonts, overriding document-specified fonts** option.
8. Click **OK**.

If this procedure does not deliver the expected results (that is, certain characters are not displayed properly), try the following procedure:

1. Select **View > Character Set > Set Default Character Set**.
2. Select **View > Character Set > Unicode (UTF-8)**.

Usage Scenarios

The following examples illustrate some of the advantages of using TeamSite in a global enterprise. Note that a branch scenario could also apply to a workarea, directory, or file operation (for example, New Branch, New Workarea, and Import File). Scenarios can also be applied to other locales.

Scenario 1

1. The TeamSite server is running in the Windows Japanese 2000 or Windows Japanese NT locale.
2. You create a branch with a Japanese name using WebDesk Pro running on Japanese Windows NT. This branch is created in the TeamSite Intelligent File System with Windows Japanese encoding.
3. You can navigate this branch with the Japanese name using WebDesk or WebDesk Pro.
4. You can also log on to the server machine and access this branch with Japanese name using the file system interface (Windows Explorer).

Scenario 2

1. The TeamSite server is running in the Windows Japanese 2000 or Windows Japanese NT locale.
2. Your TeamSite Administrator copies a directory from the Windows Explorer file system into the TeamSite Intelligent File System. This directory contains file and directory names with Japanese encoded names.
3. Your TeamSite Administrator creates a file in the TeamSite Intelligent File System with a Japanese encoded name.
4. WebDesk Pro and WebDesk users (on any client platform) can view and access this directory (and corresponding files) with a Japanese name.

Scenario 3

1. Type an `iwsubmit` command in a shell window running on a Japanese NT system.
2. Create submit comments in Japanese.
3. Execute the `iwsubmit` command. In WebDesk or WebDesk Pro the Japanese submit comments are displayed correctly with the corresponding entity submitted.

Appendix E

Client/Server Compatability

	Server OS			
	Windows 2000 US	Windows 2000 Japanese	Windows 2000 French	Windows 2000 German
Client OS				
Windows 98 French			x	
Windows 98 German				x
Windows 98 Japanese		x		
Windows 98 US	x			
Windows NT French			x	
Windows NT German				x
Windows NT Japanese		x		
Windows NT US	x			
Windows 2000 French			x	
Windows 2000 German				x
Windows 2000 Japanese		x		
Windows 2000 US	x			
MacOS 9.0 Japanese		x		
MacOS 9.0 US	x			
Solaris 2.6, 2.7, 2.8 (C locale)	x			

	Server OS			
	Windows 2000 US	Windows 2000 Japanese	Windows 2000 French	Windows 2000 German
WebDesk UI, Launchpad, and Templating UI				
French			x	
German				x
Japanese		x		
English	x	x	x	x
File, directory, store, workarea, and branch names				
French			x	
German				x
Japanese		x		
English	x	x	x	x
Templating content and metadata				
French	x	x	x	x
German	x	x	x	x
Japanese	x	x	x	x
English	x	x	x	x
Traditional Chinese	x	x	x	x
Simplified Chinese	x	x	x	x

	Server OS				
	Solaris 2.6, 2.7, 2.8 C locale	Solaris 2.7 de locale (German)	Solaris 2.7 fr locale (French)	Solaris 2.7 Shift-JIS locale (Japanese)	Solaris 2.7 EUC-JP locale (Japanese)
Client OS					
Windows 98 French			x		
Windows 98 German		x			
Windows 98 Japanese				x	x
Windows 98 US	x				
Windows NT French			x		
Windows NT German		x			
Windows NT Japanese				x	x
Windows NT US	x				
Windows 2000 French			x		
Windows 2000 German		x			
Windows 2000 Japanese				x	x
Windows 2000 US	x				
MacOS 9.0 Japanese				x	x
MacOS 9.0 US	x				
Solaris 2.6, 2.7, 2.8 (C locale)	x				

	Server OS				
	Solaris 2.6, 2.7, 2.8 C locale	Solaris 2.7 de locale (German)	Solaris 2.7 fr locale (French)	Solaris 2.7 Shift-JIS locale (Japanese)	Solaris 2.7 EUC-JP locale (Japanese)
WebDesk UI, Launchpad, and Templating UI					
French			x		
German		x			
Japanese				x	x
English	x	x	x	x	x
File, directory, store, workarea, and branch names					
French			x		
German		x			
Japanese				x	x
English	x	x	x	x	x
Templating content and metadata					
French	x	x	x	x	x
German	x	x	x	x	x
Japanese	x	x	x	x	x
English	x	x	x	x	x
Traditional Chinese	x	x	x	x	x
Simplified Chinese	x	x	x	x	x

	Server OS	
	Windows NT US	Windows NT Japanese
Client OS		
Windows 98 French		
Windows 98 German		
Windows 98 Japanese		x
Windows 98 US	x	
Windows NT French		
Windows NT German		
Windows NT Japanese		x
Windows NT US	x	
Windows 2000 French		
Windows 2000 German		
Windows 2000 Japanese		x
Windows 2000 US	x	
MacOS 9.0 Japanese		x
MacOS 9.0 US	x	
Solaris 2.6, 2.7, 2.8 (C locale)	x	
WebDesk UI, Launchpad, and Templating UI		
French		
German		
Japanese		x
English	x	x

	Server OS	
	Windows NT US	Windows NT Japanese
File, directory, store, workarea, and branch names		
French		
German		
Japanese		x
English	x	x
Templating content and metadata		
French	x	x
German	x	x
Japanese	x	x
English	x	x
Traditional Chinese	x	x
Simplified Chinese	x	x

Index

Symbols

- .uid files
 - location of 72

A

- aborting server operations 109
- absolute paths 167
- access
 - privileges, to TeamSite 69
 - to files 70

- accessing TeamSite 55
- ACEs
 - about 160
 - changing at submit time 161

- ACLs
 - about 160
 - changing 74
 - changing at submit time 158
- activating backing stores 254

- adding
 - custom menu items 125
 - metadata capture to the TeamSite GUI 209
 - metadata search to the TeamSite GUI 216
 - users to TeamSite 69, 72, 93

- administration GUI
 - about 87
 - and iw.cfg 88
 - and log files 88

- applying settings 90
- logging in 91
- navigating 89
- performing server operations 88
- refreshing 90
- viewing system information 87, 91

Administrators

- abilities 80
- about 70
- defined 18

- application variables 274

area labels

- configuring 100, 114

- assigning files 18

- attribute filtering 158

attributes

- in LDAP schemas 141
- of windows in custom menu items 129

authentication

- expiration 122
- external file 138
- LDAP 138
- password 69
- setting type 138
- user 138

Authors

- abilities 80
- about 70

- creating tasks 119
- defined 18
- editing files 119

Autoprivate

- about 148
- configuring 148
- matching filenames 149
- matching patterns 148

autoprivat.cfg

- about 111, 267
- format 148
- location 148, 267
- sample 150

B

backing store

- access control 261
- activating 254
- backing up 264
- comments 253
- converting from command line 246
- deleting 232
- disk space 26
- encoding of names 252
- freezing 155, 157
- location 28, 29
- moving 254
- multibyte characters 252
- repairing 226



- backups
 - multiple stores 264
 - of workareas 264
 - strategies 265
- branch and workarea security 97, 145
- branches
 - creating 56
 - defined 15
 - locking models on 77
 - permissions 79
 - read access 145
 - remappings, configuring 169
 - restrictions on names 57
 - setting locking model 57
 - structure 234
- browsers
 - clearing cache 111
 - compatibility with TeamSite 27
 - requirements 27
 - windows, configuring 123
- buttons, disabling 132
- C**
- cache size 104, 155
- captured subexpression 276
- Casual Contributor Interface
 - about 119
 - expiring authentication 122
- CGI scripts
 - adding to the GUI 125
 - creating 126
 - in WebDesk Pro 128
 - internationalization 312
 - window attributes 129
- CGI wrapper
 - available variables 126
 - enabling Perl scripts 125
- Change permissions
 - on directories 79
- changing
 - file attributes, on
 - submission 158
 - group ownership of
 - workareas 75
 - TeamSite file locations 232
 - TeamSite mount 232
 - valid search paths 217
- charset parameter, content
 - encoding 313
- checking
 - disk space usage 233
 - request handling 220
 - server status 220
- clients
 - TeamSite 55
- CLTs
 - code page requirements 311
 - internationalization 311
 - iwconvert 246
 - iwfreeze 250
 - iwidmap 258, 261
 - iwmigrate 259
 - iwreset 253
 - iwstoreadm 254
- code pages 311
- comments
 - submit
 - individual file 61
 - keywords 61
 - submit operation 61
- comparing files
 - viewing results 77
- compressing
 - editions 234
- configuration files
 - list of 267
 - locations 146
 - moving 146
- configuration options
 - available 111
 - requiring a restart 111
- configuring
 - area labels 100, 114
 - Autoprivate 148
 - backing store freezes 105, 155, 157
 - branch and workarea
 - security 97, 145
 - branch remappings 169
 - cache size 104, 155
 - custom menu items 125
 - different web servers 178
 - directory operations 134
 - disabling Editor publish
 - capability 97, 118
 - disk space 26
 - domain lists in the login
 - screen 122
 - domains for group
 - authentication 95, 96, 145
 - edition views 99, 116
 - encoding of text files 313
 - external remappings 179
 - file locations 146
 - file system active area
 - cache 156
 - file system threadcount 105, 156
 - history views 99, 117
 - Home pages 117
 - IP addresses 225
 - iw-mount alias 41
 - job attributes 136

- jobs in the GUI 99, 135
 - launching files through
 - iwproxy 153
 - LDAP 138, 140
 - lock behavior 98, 144
 - log files 105
 - main branch locking
 - model 143
 - main branch ownership 144
 - main configuration file 111
 - menu items 132
 - metadata capture 190
 - new browser windows 123
 - preview windows 99, 123
 - proxy server 166
 - RPC threadcount 105, 156
 - rule sets for metadata
 - capture 195
 - Submit and Update logs 144
 - Submit button 131
 - submit filtering 158
 - TeamSite administration
 - GUI 87
 - templates 151
 - throughput monitors 105, 157
 - web servers 40, 100
 - conflicting edits 77, 78
 - conserving disk space 234
 - control panel, Regional
 - Setting 154
 - conventions
 - path name 13
 - conventions, notation 11
 - copying files 60
 - CPU requirements 24
 - creating
 - branches 56
 - tasks, through WebDesk 119
 - workareas 59, 70
 - custom menu items
 - adding to the GUI 125
 - CGI scripts 125, 128
 - HTML pages 130
 - window attributes 129
 - custom scripts
 - creating 126
 - enabling 125
- D**
- data store 232
 - database, LDAP 140
 - datacapture.cfg
 - about 188
 - annotated example
 - database element 202
 - DATE datatype 203
 - instance 203
 - metadata identifier 202
 - rule identifier 202
 - UTF-8 encoding 202
 - validation-regex 203
 - configuring 195
 - DTD 196
 - location of 188
 - debugging
 - proxy server configuration 183
 - submit filtering 163
 - default
 - code pages 311
 - default file locations 28
 - deleting
 - backing store 232
 - branches 236
 - directories
 - disabling operations on 134
 - permissions 79
 - shared 232
 - disabling
 - buttons 132
 - directory operations 134
 - menu items 132
 - SmartContext Editing 118
 - Submit button 131
 - unlocked file upload 98, 135
 - disk space
 - checking usage 233
 - compression 234
 - conserving 234
 - data store 232
 - file system mount 233
 - low 157
 - managing 232
 - moving the backing store 236
 - partitions 26
 - recovery 234
 - removing old versions 236
 - requirements 24
 - document root
 - configuring 169
 - defined 168
 - mapping 168
 - domains
 - configuring, in the login
 - screen 95, 122
 - for group authentication 145
 - DTD
 - datacapture.cfg 196
 - metadata-rules.cfg 191
- E**
- editing
 - files through WebDesk 119
 - text editor application 27
 - editions

- allowing Editors to publish 118
- defined 17
- initial 57
- new 63
- number of displayed 116
- publishing 63
- see also* publishing editions
- viewing 99, 116
- views, configuring 99, 116
- Editors
 - abilities 80
 - about 70
 - defined 18
 - disabling publish capability 97, 118
- email
 - and tasks 137
 - mapping files 137
 - setting domains 137
 - setting servers 137
 - settings, for workflow 137
- Embedded Failsafe 184
- enabling
 - SmartContext Editing 118
 - SmartContext QA 153
- encoding
 - charset parameter 313
 - file_encoding.cfg 283, 313
 - html files 269
 - IANA preferred names 283
 - Merge tool 283
 - META tag 313
 - of backing store names 252
 - of contents of iw.cfg 252
 - setting in iw.cfg 138
 - Single Byte Character Sets 311
 - Source Differencing tool 283
 - specifying 313
 - text editors 253, 314
 - text files 269
 - Unicode 310
 - UTF-8 282, 310
 - valid charsets 283
 - visual differencing 283
 - vpaths 269
- entity database 117
- errata 13
- external remappings, configuring 179
- F**
 - failover *see* proxy server
 - file attributes
 - changing on submission 158
 - file system
 - active area cache 156
 - mount 233
 - performance, improving 232
 - threadcount 105, 156
 - file system interface
 - network connection 28
 - using 28
 - file_encoding.cfg 111, 269, 313
 - Merge tool 283
 - sample file 284
 - SmartContext Editing 282
 - Source Differencing tool 283
 - UTF-8 282
 - valid encodings 283
 - visual differencing 283
 - files
 - access to 70
 - assigning 18
 - configuration 111, 269
 - creating 151
 - encoding 111, 269
 - file_encoding.cfg 111, 269, 283, 313
 - merging
 - and Submit locking 77
 - and Write locking 78
 - permissions 79
 - private 148
 - setting permissions on 70
 - submitting to the staging area 61
 - templates 151
 - virtual 234
 - filtering, on file submission 158
 - finding installation directory 221
 - freezing the backing store 109, 155, 157
 - fully-qualified paths
 - see* proxy server
- G**
 - generating new encryption keys 226
 - Global Report Center
 - requirements 26
 - group authentication
 - domains 95, 96, 145
 - groups
 - creating 75
 - files 73
 - membership 75
- H**
 - hardware requirements 24
 - High Availability
 - about 287
 - components 288
 - configuring 287, 290
 - installing 290

- iw.powerfail 289, 290
- iw.processfail 289, 292
- iwtock 289
- logging 290, 292
- starting and stopping the
 - server 294
- uninstalling 294
- history views
 - configuring 99, 117
- Home page
 - setting 117
- host headers, remapping
 - see also* proxy server
- host permissions
 - setting 95
- HTML pages
 - adding to the GUI 130
- HTTP port number 30, 38
- httpd user name 41
- HTTPS requests
 - redirecting 52

I

- IANA charset names 283
- IIS
 - and the server mount 221
 - auto configuration script 40
 - configuring 40
- in-context QA
 - enabling 153
- initial edition 57
- installation directory
 - locating 221
- installing
 - license keys 33
 - required access for 29
 - TeamSite 29
 - TeamSite Templating 35

- Intelligent File System
 - about 21
- intermediate variables
 - variables
 - intermediate 274
- internationalization
 - browser behavior 314
 - CGI scripts 312
 - CLTs 311
 - encoding setting in iw.cfg 138
 - file_encoding.cfg 269
 - IANA charset names 283
 - iw.cfg settings 154
 - recommendations 313
 - server_locale setting 154
 - SmartContext Editing 269
 - text editor encoding 252, 314
 - Unicode 310
 - UTF-8 282, 310
 - vpath encoding 269
- Internet Explorer
 - compatibility 27
- Interwoven administration GUI
 - see* administration GUI
- Interwoven Merge tool 269
- invalidating user sessions 226
- IP addresses
 - changing 225
- iPlanet web server
 - configuring 40
 - configuring aliases 53
- iw.cfg
 - about 111
 - activating change to 253
 - and the administration GUI 88
 - and the proxy server 168, 169
 - configuration options 111
 - encoding of 252

- encoding setting 138
- locating 34, 268
- metadata search paths in 217
- specifying file locations 232
- iwchgrp 75
- iwckrole 76
- iwconvert 246
- iwfreeze 250
- iwgetelog 223
- iwgetlocation 33
- iw-home
 - about 28
- iwidmap 258, 261
- iwmigrate 259
- iw-mount alias
 - configuring 41
- iwprefconv 117
- iwproxy
 - configuring 166
 - debug option 183
 - launching files 153
- iwreset 253
- iwserver
 - checking number of 220
 - locating 221
 - memory usage 220
- iwsessionkeygen 226
- iwstat 223
- iw-store directory
 - backing up 36
- iwstoreadm 254
- iwtemplates.cfg
 - about 111
 - format 151
 - location 151, 267

J

- Java

- servlet engine 100, 143
- JavaScript 53
- jobs
 - attributes 136
 - defined 20
 - filters 136
 - listing in the GUI 99, 135
 - settings 136

L

- labels, configuring 114
- languages
 - browser behavior when interpreting encoding 314

- LaunchPad
 - about 54
 - applet 121
 - application 121
 - installing 54
 - localized GUI 311
 - setting server names 121
 - setting the default interface 121

- LDAP
 - and OpenAPI 139, 141
 - and operating system authentication 142
 - configuring 138, 140
 - database 140
 - modifying schemas 141
 - schemas 139, 140
 - user and role authentication 138
 - user authentication 138

- license keys
 - generation page 32
 - installing 33
 - obtaining 31

- troubleshooting 34
- loading content 56
- local domains 28
- localized features 307
- locales
 - native 154
 - TeamSite server setting 154

- locating
 - installation directory 221

- locations
 - of configuration files 146
 - of iw.cfg 34, 268
 - of roles files 268
 - of TeamSite files 146
 - changing 232

- locking
 - files, and uploading 98, 135
 - in workareas 77
 - Mandatory Write, defined 78
 - Optional Write, defined 78
 - Submit, defined 77
 - types of 77
 - Write, about 78

- locking model
 - on the main branch 143
 - setting 57

- locks
 - configuring behavior of 98, 144
- log files
 - and the administration GUI 88
 - configuring 105
 - reviewing 222
 - viewing 107
 - through the GUI 88

- logging
 - users and groups 146
- logging in authentication 69

- to the administration GUI 91
- logging users out 226
- login authentication expiration
 - default 122
 - setting 122
- login names 72
- login screen
 - configuring domain lists 122
 - selecting a GUI 53
 - selecting role 53
- logs
 - submit 144
 - update 144
- low disk space
 - detecting 105, 157

M

- main branch
 - locking model 143
 - ownership 144
- managing server resources. *see* server resources
- Mandatory Write locking
 - defined 78
- Masters
 - abilities 80
 - about 19, 70
- memory requirements 25
- menu items
 - disabling 132
- Merge tool 269, 283
- merging files
 - and Submit locking 77
 - and Write locking 78
- META tag
 - specifying web asset encoding 313
- metadata capture

- about 187, 188
- adding to the TeamSite
 - GUI 209
- and DAS
 - synchronizing 215
- and workflow 211
- components 188
- configuration files 188
- configuring 190
 - appearance 188
 - names 188
- DTD 191, 195
- extended attributes 210
- form 195
- initiating from within a job 211
- required input 189
- results of 210
- rule sets 195
- schematic 189
- validating input 189
- metadata search
 - about 187, 212
 - adding to the TeamSite
 - GUI 216
 - and DataDeploy 213
 - and iw.cfg 214
 - and metadata capture 212
 - changing valid search paths 217
 - components 213
 - configuration files 213
 - configuring 215
 - making fields non-
 - searchable 217
 - overview 212
 - prerequisites 212
- metadata-rules.cfg
 - about 188
 - configuring 191

- DTD 191
- examples 191
- location of 188
- rule identifier 192
- UTF-8 encoding 192
- vpath identifier 192
- monitoring
 - system status 157
- mounting the TeamSite
 - server 221
- moving
 - configuration files 146
- moving backing stores 254
- multibyte characters
 - browser behavior when
 - interpreting encoding 314
 - in backing store names 252
- MultiStore
 - backing up 264
 - defined 15, 237

N

- navigating through the
 - administration GUI 89
- Netscape browser
 - compatibility 27
- Netscape Enterprise Server
 - configuring aliases 53
- network drive 55
- new users
 - adding 69
- notation conventions 11
- Notepad
 - saving documents as
 - UTF-8 314

O

- OpenAPI

- and LDAP 139, 141
- OpenAPI server
 - about 224
 - starting and stopping 225
 - verifying 224
- Optional Write locking
 - defined 78
- ownership
 - of workareas, changing 75
- P**
- passwords
 - authentication 69
- path name conventions 13
- paths
 - absolute 167
 - relative 167
 - resolving *see also* proxy server
- performance
 - monitoring 157
- permissions
 - branch 69, 79
 - directory 79
 - file 69, 70, 79
 - required for actions 78
 - types of 78
 - workarea 69, 79
- port 80 30, 38
- port number
 - http server 30, 38
 - proxy server 166
 - servlet 143
 - web server 30, 39, 166
 - specifying 41
- preview windows
 - configuring number 99, 123
- printing system information 92
- private files 148

- Professional Services 226
- profiles, user 117
- program files
 - location 28, 29
- proxy server
 - about 164
 - configuring 164, 166
 - basic operation 166
 - mappings 101
 - through the GUI 100
 - to use different
 - webservers 178
 - debugging 183
 - document roots 170
 - external remappings 179
 - failover 181
 - fully-qualified paths 171
 - client configuration 172
 - configuring Internet Explorer 175
 - configuring Netscape 172
 - server configuration 171
 - host header remappings 180
 - host name 166
 - port number 166
 - redirecting TeamSite
 - views 175, 176, 178
 - relative and absolute paths 167
 - remapping document roots
 - rules of precedence 164
 - SSI remapping 181
 - publishing editions
 - about 63
 - Editors' ability to 18, 118
 - first edition 63
 - through the command line 63
 - through the TeamSite GUI 63

R

- RAID 0+1 26
- reconfiguring IP address 225
- recovering disk space 234
- redirecting HTTPS requests 52
- redirecting TeamSite views *see also* proxy server
- redirecting TeamSite views *see* proxy server
- redirector module
 - and iwproxy 153
 - for SSIs, installing 47
- regex_map
 - element 272
 - illustrated 270
 - internationalization
 - internationalization regex_map 282
 - introduced 269
 - regular expression syntax 273
 - UTF-8 282
 - variables 273
- regex_map element 269
- Regional Settings control
 - panel 154
- regular expressions
 - about 11, 159
 - case-sensitivity 273
 - expression engine 273
 - file encoding 269
 - in New File templates 152
 - in regex_maps 273
 - in Submit filters 159
- relative paths
 - about 167
 - see also* proxy server
- remote contributors 164
- removing

- menu items 132
- temp_workareas 251
- users 72, 73, 93
- repairing backing store 226
- request handling
 - checking 220
- requirements
 - backing store 26
 - client 27
 - CPU 24
 - disk space 24, 26
 - hardware 24
 - memory 25
- resetting the TeamSite server 110
- resolving path names *see also* proxy server
- reviewing TeamSite logs
 - overview 222
 - with log files 223
 - with Windows NT Event Viewer 222
- roles
 - about 70
 - adding users 93
 - TeamSite 69
- roles files
 - adding users to 72
 - locating 268
 - location of 72
 - master users needed 41
- RPC threadcount
 - configuring 105, 156
- rule sets
 - configuring 195

S

- SCE
 - see* Smart Context Editing

- ul style="list-style-type: none;">
- schemas, LDAP 139, 140
- SCSI controllers and drives 26
- searching metadata
 - components 213
 - configuring 215
 - overview 212
 - prerequisites 212
- second-predecessor links 250
- security 69
- server
 - load, monitoring 223
 - mount errors 221
 - mount, verifying 221
 - names, setting, for
 - LaunchPad 121
 - operation, verifying 220
 - operations, aborting 109
 - operations, through the
 - administration GUI 88, 109
 - resources
 - disk space 232
 - managing 232
 - shared directories 232
 - status, checking 220
- server locales 154
- server_locale setting 154
- servers
 - starting and stopping 225
- server-side includes 46
- Service Packs, uninstalling 64
- servlet engine 100, 143
- servlet port 143
- SetHomePage feature 117
- setting
 - Home pages 117
 - TeamSite permissions 96
- settings
 - email, for workflow 137
 - encoding 138
 - server_locale 154
 - System Locale 154
- shared directories 232
- shared volume 28, 55
- Single Byte Character Set (SBCS) 311
- SmartContext Editing
 - disabling 118
 - enabling 118
 - encoding of text files 269
 - file_encoding.cfg 282
- SmartContext QA
 - enabling 153
 - for SSIs 46
- Source Differencing tool 269, 283
- SSIs
 - enabling 46
 - requests, remapping and virtualizing *see also* proxy server
- SSL support 164
- staging area
 - defined 16
 - on a new branch 57
- starting TeamSite server 223
- startup time
 - reducing 145
- stopping TeamSite server 223
- Submit and Update logs, size of 144
- Submit button
 - configuring 131
 - disabling 131
- submit filtering
 - about 158
 - debugging 163
 - sequence of events 161
 - when populating a workarea 60
- Submit locking
 - defined 77
 - submitting files under 77
- submit workflow process and the Submit button 131
- submit.cfg
 - about 111, 158
 - format of 158
 - location 158
 - sample 161
- submitting
 - files, to the staging area 61
 - locked files 144
 - under Submit locking 77
 - under Write locking 78
- submitting files, changing attributes 158
- synchronizing metadata capture and DAS 215
- system information
 - printing 92
 - viewing 87, 91
- System Locale setting 154
- system services
 - iwprefconv 117
- system status, monitoring 157
- ## T
- task ownership 79
 - tasks
 - and email 137
 - defined 20
 - TeamSite
 - accessing
 - through the file system 55
 - through the GUI 53

- adding metadata capture to the GUI 209
- adding metadata search to the GUI 216
- administration GUI 87
- architecture 21
- clients 53, 55
- configuration files 267
- configuring
 - through the GUI 87
- file locations
 - changing 146
- file locations, changing 232
- High Availability 287
 - see also* High Availability
- installing 29
- mount, changing 232
- mounting 55
- permissions, setting 96
- populating 60
- proxy server 164
 - configuring mappings 101
- proxy server *see also* proxy server
- proxy server, configuring 100
- resetting the server 110
- roles 69, 93
- uninstalling 65
- uninstalling Service Pack 1 64
- upgrading 35
- user roles 70, 93
- users, adding and removing 93
- web daemon 164
- TeamSite installation
 - CD-ROM 29, 36
- TeamSite server
 - answering requests 220
 - enhancing file system
 - performance 232
 - freezing and unfreezing 109
 - mounting 221
 - starting 223
 - stopping 223
 - TeamSite Templating
 - installing 35
 - localized GUI 311
 - Teamsite.exe 29, 36, 37
 - Technical Support 226
 - temp_workareas 251
 - templates
 - configuring 151
 - text editor encodings 253, 314
 - encoding
 - text editors 282
 - throughput monitors 105, 157
 - troubleshooting
 - license keys 34
 - overview 226
 - repairing backing store 226
 - TeamSite access 226
 - tslicinfo.log file 32
- U**
- Unicode
 - about 310
- uninstalling TeamSite 64, 65
- unlocked file upload,
 - disabling 98, 135
- upgrading TeamSite 35
- uploading files 98, 135
- URL access 119
- user authentication 138
 - re-encrypting 226
- user profiles 117
 - about 117
- user roles
 - about 70
 - Administrator 70
 - Author 70
 - checking 76
 - Editor 70
 - Master 70
 - permitted actions 79, 80
- user sessions
 - invalidating 226
- users
 - adding 72
 - expiring authentication 122
 - removing 72, 73
- users and groups
 - logging 146
- UTF-8
 - about 310
 - file_encoding.cfg 282
 - recommendations 313
 - regex_map 282
- V**
- valid search paths
 - for metadata 217
- variables
 - application 274
 - captured subexpression 276
 - naming convention 274
- variables, regex_map 273
- verifying server mount 221
- versions
 - number of displayed 117
- viewing
 - branches and workareas 145
 - log files 107
 - system information 91
- virtual files 234

- vpaths
 - encoding mappings 269

W

- web browsers
 - behavior when interpreting
 - encoding 314
 - requirements 27
- web content
 - specifying the encoding of 313
- web daemon
 - about 164
 - configuring 164, 166
 - setting defaults 142
- web server
 - host name 166
 - port number 41, 166
- web servers
 - aliases 28
 - configuring 40, 100, 178
 - group 95, 96, 142
 - iPlanet
 - configuring aliases 53
 - Netscape
 - configuring aliases 53
 - plugins 153, 166
 - port number 30, 39
 - starting and stopping 166
 - stopping and starting 51
- WebDesk
 - about 53
 - expiring user
 - authentication 122
 - localized GUI 311
- WebDesk Pro
 - about 53
 - adding custom menu items 128, 130

- window attributes
 - for custom menu items 129
- Windows NT
 - Event Viewer 222
 - Task Manager 220
 - User Manager 72
- Windows permissions
 - interaction with TeamSite 79
- Wordpad
 - saving documents as
 - UTF-8 314
- workareas
 - changing group ownership 75
 - creating 59, 70
 - defined 16
 - locking files in 77
 - permissions 79
 - populating 60
 - read access 145
 - submitting to the staging area 61
 - temp_workareas 251
- workflow
 - and metadata capture 211
 - jobs, defined 20
 - models
 - and jobs 20
 - defined 19
 - tasks
 - defined 20
 - specifying files in 211
 - troubleshooting 41
- Write locking
 - about 78
 - submitting files 78
 - see also* Optional Write locking, Mandatory Write locking

X

- XML
 - about 11
 - datacapture.cfg 196
 - metadata-rules.cfg 191
 - regex_map language 269
 - special characters 279